

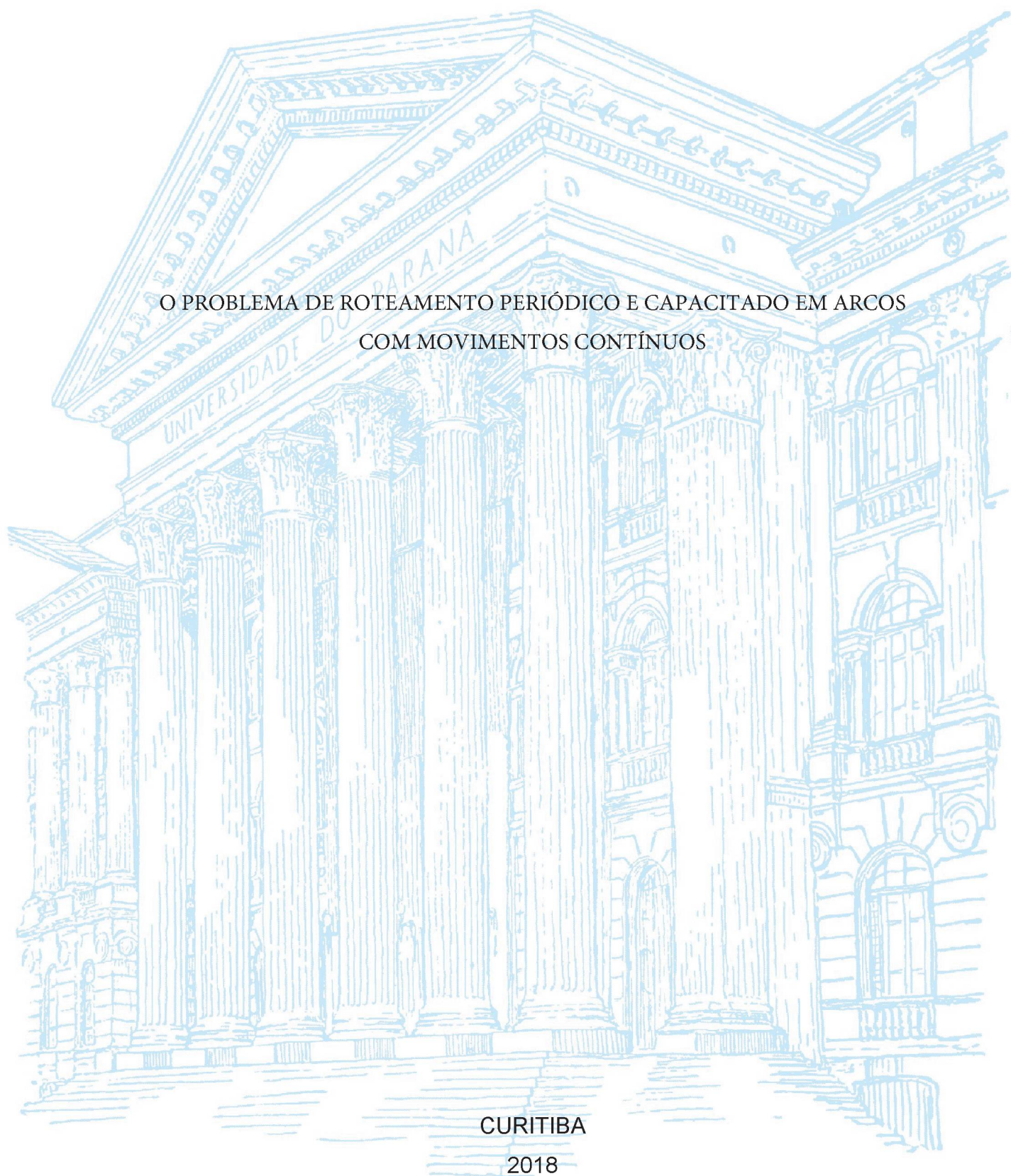
UNIVERSIDADE FEDERAL DO PARANÁ

GUILHERME VINICYUS BATISTA

O PROBLEMA DE ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS
COM MOVIMENTOS CONTÍNUOS

CURITIBA

2018



GUILHERME VINICYUS BATISTA

O PROBLEMA DE ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS
COM MOVIMENTOS CONTÍNUOS

Tese apresentada ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática, Departamentos de Construção Civil e de Matemática, Setores de Tecnologia e de Ciências Exatas, Universidade Federal do Paraná, como parte das exigências para a obtenção do título de Doutor em Métodos Numéricos em Engenharia.

Orientador: Prof. Dr. Cassius Tadeu Scarpin

Coorientador: Prof. Dr. José Eduardo Pécora Junior

CURITIBA
2018

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

B333p

Batista, Guilherme Vinicyus

O problema de roteamento periódico e capacitado em arcos com movimentos contínuos [recurso eletrônico] / Guilherme Vinicyus Batista. – Curitiba, 2018.

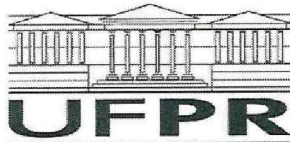
Tese - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2018.

Orientador: Cassius Tadeu Scarpin – Coorientador: José Eduardo Pécora Junior.

1. Arcos. 2. Ferrovias – Construções e estruturas. 3 Ferrovias – Trilhos. 4. Medidas de segurança. 5. Ferrovias – Manutenção e reparos. 6. Colônia de insetos. 7. Formigas. I. Universidade Federal do Paraná. II. Scarpin, Cassius Tadeu. III. Pécora Junior, José Eduardo. IV. Título.

CDD: 625

Bibliotecário: Elias Barbosa da Silva CRB-9/1894



MINISTÉRIO DA EDUCAÇÃO
SETOR CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO MÉTODOS NUMÉRICOS
EM ENGENHARIA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MÉTODOS NUMÉRICOS EM ENGENHARIA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **GUILHERME VINICYUS BATISTA** intitulada: **O Problema de Roteamento Periódico e Capacitado em Arcos com Movimentos Contínuos**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 22 de Outubro de 2018.

CASSIUS TADEU SCARPIN
Presidente da Banca Examinadora

VOLMIR EUGÊNIO WILHELM
Avaliador Interno

GUSTAVO VALENTIM LOCH
Avaliador Interno

LEANDRO MAGATÃO
Avaliador Externo

MARIANA KLEINA
Avaliador Externo

Dedico esse trabalho a minha esposa Ana Júlia Lucht Rodrigues e a todos meus familiares.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelas oportunidades que tive e por ter conhecido tantas pessoas maravilhosas ao longo desse caminho. Sem dúvida, se não fosse algumas pessoas, eu não teria chegado a esse ponto marcante que é a finalização de um doutorado.

Agradeço a minha esposa, Ana Júlia, que me apoiou incondicionalmente ao longo de todo o período de mestrado e doutorado. Esteve sempre ao meu lado nos momentos felizes e nos difíceis e tristes, me ajudou a levantar e seguir em frente, além de ter se doado nos momentos que mais precisei.

Agradeço ao meu pai, Mauro Cesar Batista, e a minha mãe, Rosana Schwarz Batista, por todos os esforços que fizeram para que eu tivesse uma boa educação. Não tenho palavras para descrever o tamanho de minha gratidão. Cada vez mais percebo o tamanho da dedicação que tiveram na árdua tarefa de preparar os filhos para o mundo. Obrigado por terem sido meu porto seguro e por tudo que fizeram e ainda fazem por mim. Agradeço também aos meus irmãos, cunhados, sogro, sogra e afilhada por todo apoio e força.

Agradeço ao meu orientador e amigo Dr. Cassius Tadeu Scarpin, o qual tive a sorte de ter como professor e como colega na graduação. Durante todos esses anos de convivência minha admiração por sua forma de trabalho, dedicação, comprometimento e caráter só cresceu, tenho-o como exemplo a ser seguido. Obrigado pelas inúmeras vezes que me motivou a persistir nesse caminho, obrigado por estar ao meu lado em momentos importantes da minha vida e ajudando na realização de alguns sonhos.

Agradeço ao meu coorientador Dr. José Eduardo Pécora Jr. por toda a disponibilidade e pelo voto de confiança logo no início do doutorado. Mesmo sem me conhecer aceitou a tarefa de me coorientar dando contribuições valiosas para o desenvolvimento dessa pesquisa, além da ajuda na viabilização do meu intercâmbio para o Canadá. Aproveito esse momento também para agradecer ao meu orientador durante o período no exterior, Dr. Angel Ruiz, que me recebeu de portas abertas na Université Laval em Quebec, foi um período de muito crescimento. Aos dois, obrigado pela parceria e carinho no desenvolvimento de nossas pesquisas, admiro vocês.

Agradeço a todos os professores que ajudaram na minha formação, foram muitos os que marcaram e que contribuíram. Tive a sorte de receber dicas de alguns deles em momentos difíceis, como a escolha do curso de graduação, e também sobre o que fazer após a graduação. Gostaria de agradecer também aos professores que compuseram a banca que avaliou este trabalho Dr. Gustavo Valentim Loch, Dr. Leandro Magatão, Dr. Volmir Eugênio Wilhelm e Dra. Mariana Kleina, suas contribuições foram valiosas e fico feliz em saber que vocês acompanharam o desenvolvimento desse trabalho.

Agradeço aos membros do Grupo de Tecnologia Aplicada à Otimização (GTAO). Foram muitos os momentos juntos, sempre com muito aprendizado e ajuda mútua entre os membros. Tenho um carinho de irmão para com todos, tive o prazer de ser um dos primeiros membros e vi muitos chegarem, para mim é nítido que juntos somos mais. Agradeço também a todos os amigos externos a universidade, obrigado por todos os momentos que vivemos e pelo apoio.

Agradeço a Universidade Federal do Paraná (UFPR), me tornei aluno em 2004 quando ingressei no Ensino Médio da Escola Técnica, passei pelo curso de Engenharia de Produção e, finalmente, fiz mestrado e doutorado no Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE). Tenho orgulho de ser UFPR. Sem o ensino público e de qualidade que tive nessa instituição não sei o que seria. A UFPR me abriu muitas portas e serei eternamente grato. Agradeço ao PPGMNE pelo suporte e estrutura, ao secretário Jair Anjos que esteve sempre disponível e pronto para auxiliar nas tarefas do programa. Agradeço também a CAPES Reuni pelo apoio financeiro.

“Todo caminho construído é um acúmulo de persistência. Os caminhos levam o homem para fora de si mesmo, a fim de ele ir à procura de alguma coisa, além dos limites comuns de sua própria existência.”

Relatório da expedição Estrada do Sol, Peru, 1948

RESUMO

Esse trabalho aborda o Problema de Roteamento Periódico e Capacitado em Arcos, *Periodic Capacitated Arc Routing Problem* (PCARP), que pode ser definido como um problema de roteamento em arcos envolvendo um horizonte de tempo composto por mais de um período, os quais podem ser formados por qualquer fração temporal desde que discretas. Um ou mais veículos, ao fazerem suas rotas, devem executar da melhor maneira possível as tarefas que estão associadas a cada arco sem exceder sua capacidade e respeitando as frequências exigidas. Esse trabalho visa classificar as pesquisas acerca do PCARP por meio de uma proposta taxonômica que considera características físicas, foco aplicável ou teórico e abordagem resolutiva. Outra contribuição é um estudo aplicado à inspeção de ferrovias, o qual é classificado como PCARP com Movimentos Contínuos. Rotas com movimentos contínuos são caracterizadas por não terem a obrigatoriedade de serem iniciadas e terminadas em um depósito. O problema tem sua formulação matemática apresentada e sua resolução é feita utilizando métodos exato, heurístico e híbridos. O método exato consiste na utilização de um *solver* comercial e a heurística utilizada é um novo algoritmo de Otimização de Colônia de Formigas, *Ant Colony Optimization* (ACO). Ambos têm seus pontos fortes que se complementam, sendo assim foram propostas três abordagens híbridas. A primeira consiste na utilização da colônia de formigas como solução inicial para o modelo exato, e as outras duas abordagens são variações do algoritmo Busca Iterativa em Espaço Restrito, *Iterative Restricted Space Search* (IRSS). A IRSS é baseada na exploração estratégica do espaço de soluções de um dado problema que, nesse trabalho, será auxiliada pelo algoritmo ACO. Os resultados alcançados mostram que as estratégias de resolução propostas são capazes de superar os resultados alcançados pelo *solver* comercial.

Palavras-chave: Problema de Roteamento Periódico e Capacitado em Arcos 1. Movimentos Contínuos 2. Otimização da Colônia de Formigas 3. Busca Iterativa em Espaço Restrito 4. Manutenção e Inspeção de Ferrovias 5.

ABSTRACT

The Periodic Capacitated Arc Routing Problem (PCARP) can be defined as an arc routing problem that involves a time horizon, which is greater than one period that can be composed by any fraction of time. One or many vehicles, while into routes, must perform, in the best way possible, all tasks in the arcs without exceeding their capacities and attending the required frequencies. This work classifies researches about PCARP using a taxonomy that considers physical features, applied or academic focus and resolution approach. An additional contribution is a study based on inspection of railroads; this one is classified as PCARP with Continuous Moves. Routes with continuous moves do not need a depot as a starting or finishing point. This work presents the mathematical formulation for the problem and tackles it with exact, heuristic and hybrid methods. The exact approach consists of the use of a commercial solver, and the heuristic is a new Ant Colony Optimization (ACO) algorithm. Both have strengths that may be combined to compose hybrid approaches, three of them are explored. The first one uses the ACO to form an initial solution to the solver, and the two others are variants of the Iterative Restricted Space Search (IRSS) framework. The IRSS strategically exploits the solution space of one problem assisted by an algorithm, in this case, the ACO. The results achieved showed that the strategies proposed are able to overcome the ones from the commercial solver.

Keywords: Periodic Capacitated Arc Routing Problem 1. Continuous Moves 2. Ant Colony Optimization 3. Iterative Restricted Space Search 4. Inspection and Maintenance in Railways 5.

LISTA DE FIGURAS

FIGURA 2.1 - UMA SIMPLES CLASSIFICAÇÃO DO PCARP	45
FIGURA 2.2 - CLASSIFICAÇÃO DE PROBLEMAS DE ROTEAMENTO EM ARCOS PERIÓDICOS.....	48
FIGURA 2.3 - TIPOS DE ROTEIRIZAÇÃO USADOS NO PCARP	49
FIGURA 3.1 - EXEMPLO HIPOTÉTICO DE PCARP-CM	63
FIGURA 3.2 - PROBLEMA DE DETERMINAÇÃO DO HORIZONTE DE TEMPO	64
FIGURA 4.1 - UM EXEMPLO DE PROBLEMA (a) E SUA REPRESENTAÇÃO NO GRAFO (b).....	77
FIGURA 5.1 - PROCESSO DO IRSS.....	98
FIGURA 5.2 - UM PEQUENO PROBLEMA COM TODAS SUAS SOLUÇÕES SIMÉTRICAS	103
FIGURA 5.3 - EXEMPLOS DE SOLUÇÕES SIMÉTRICAS QUE PODEM SER EVITADAS EM UM PROBLEMA COM 3 CARROS	105

LISTA DE GRÁFICOS

GRÁFICO 2.1 - PUBLICAÇÕES POR ANO	57
GRÁFICO 2.2 - PUBLICAÇÕES POR AUTOR	58
GRÁFICO 2.3 - REVISTAS QUE MAIS PUBLICARAM SOBRE O PCARP	58
GRÁFICO 2.4 - RELAÇÃO DE PUBLICAÇÃO POR PAÍSES	59
GRÁFICO 5.1 - PORCENTAGENS UTILIZADAS NO IRSS QUE RETORNARAM VALOR ÓTIMO	112
GRÁFICO 5.2 - QUANTIDADE DE MELHORES SOLUÇÕES ALCANÇADAS POR ITERAÇÃO COM O IRSS HEURÍSTICO	118
GRÁFICO 6.1 - ANÁLISE DAS CINCO ABORDAGENS EM NÚMEROS	123

LISTA DE QUADROS

QUADRO 2.1 - TRABALHOS ACADÊMICOS ACERCA DO PCARP.	43
QUADRO 2.2 - RESUMO DA PROPOSTA TAXONÔMICA.....	55
QUADRO 2.3 - CLASSIFICAÇÃO TAXONÔMICA DOS TRABALHOS	56
QUADRO 3.1 - PARÂMETROS DE ENTRADA DO MODELO MATEMÁTICO.....	66

LISTA DE TABELAS

TABELA 3.1 - RELAÇÃO DOS 10 PROBLEMAS GERADOS, PROPRIEDADES E RESULTADOS.....	68
TABELA 3.2 - RESULTADOS DOS 10 PROBLEMAS <i>gvb</i> PARA O MODELO 201869	
TABELA 3.3 - RESULTADOS DAS 23 INSTÂNCIAS <i>pgdbcm</i>	70
TABELA 3.4 - RESULTADO <i>pgdbcm</i> COMPARANDO MODELOS.....	71
TABELA 3.5 - NOVAS 23 INSTÂNCIAS <i>pgdbcm</i>	73
TABELA 3.6 - RESULTADOS DAS NOVAS INSTÂNCIAS PGDCM.....	74
TABELA 4.1 - RESULTADOS PARA DEFINIÇÃO DOS PARÂMETROS DO ACO ..	84
TABELA 4.2 - COMPARAÇÃO <i>SOLVER</i> VERSUS ACO EM 1 HORA.....	86
TABELA 4.3 - PERFORMANCE <i>SOLVER</i> E ACO SEGUNDO TAXA DE OCUPAÇÃO MÍNIMA	88
TABELA 4.4 - RESULTADOS DO <i>SOLVER</i> UTILIZANDO SOLUÇÃO INICIAL DADA PELO ALGORITMO ACO	89
TABELA 5.1 - RESPOSTAS DO IRSS EXATO PARA OS 39 <i>pgdbcm</i>	109
TABELA 5.2 - DETALHES DA EXPLORAÇÃO DOS RS PARA AS 39 INSTÂNCIAS	111
TABELA 5.3 - RESPOSTAS DO IRSS HEURÍSTICO PARA OS 39 <i>pgdbcm</i>	117
TABELA 5.4 - RESPOSTAS DO IRSS HEURÍSTICO PARA AS INSTÂNCIAS <i>pgdbcm22</i> E <i>pgdbcm38</i> EM UM WORKSATION.....	119
TABELA 6.1 - RESULTADOS GERAIS DO PCARP-CM INSTÂNCIAS <i>pgdbcm1</i> A <i>pgdbcm20</i>	121
TABELA 6.2 - RESULTADOS GERAIS DO PCARP-CM INSTÂNCIAS <i>pgdbcm21</i> A <i>pgdbcm39</i>	122

LISTA DE ABREVIATURAS OU SIGLAS

AM	- Algoritmo Memético
AMP	- Algoritmo Memético Periódico
ALNS	- <i>Adaptative Large Neighborhood Search</i>
CARP	- <i>Capacitated Arc Routing Problem</i>
CPP	- <i>Chinese Postman Problem</i>
GRASP	- <i>Greedy Randomized Adaptive Search Procedure</i>
HMI	- Heurística de Melhor Inserção
IRP	- <i>Inventory Routing Problem</i>
IRSS	- <i>Iterative Restricted Space Search</i>
LRP	- <i>Location Routing Problem</i>
PCARP	- <i>Periodic Capacitated Arc Routing Problem</i>
PCARP-CM	- <i>Periodic Capacitated Arc Routing Problem with Continuous Moves</i>
PCARP-IC	- <i>Periodic Capacitated Arc Routing Problem with Inventory Constraints</i>
PCARP-IS	- <i>Periodic Capacitated Arc Routing Problem with Irregular Services</i>
PDP	- <i>Pick-up and Delivery Problem</i>
PPSP	- <i>Pulp Production Scheduling Problem</i>
PRPV	- Problema de Roteamento Periódico de Veículos

SUMÁRIO

1 INTRODUÇÃO	18
1.1 OBJETIVOS	22
1.1.1 Objetivo geral	22
1.1.2 Objetivos específicos.....	22
1.2 CONTRIBUIÇÕES E LIMITAÇÕES	23
1.3 ESTRUTURA DO TRABALHO.....	24
2 PCARP: REVISÃO DE LITERATURA E PROPOSTA TAXONÔMICA.....	26
2.1 O ROTEAMENTO EM ARCOS	26
2.2 O PROBLEMA DE ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS	29
2.2.1 Contribuições dos trabalhos científicos acerca do PCARP	32
2.2.2 Proposta taxonômica para o PCARP	42
2.2.2.1 Período.....	43
2.2.2.2 Horizonte de tempo	45
2.2.2.3 Serviços.....	47
2.2.2.4 Roteirização.....	48
2.2.2.5 Dimensão da frota	49
2.2.2.6 Frequências.....	50
2.2.2.7 Demanda	52
2.2.2.8 Rigor da demanda	52
2.2.2.9 Grafo.....	53
2.2.2.10 Foco.....	53
2.2.2.11 Abordagem de resolução.....	54
2.2.3 Resumo da Proposta Taxonômica	55
2.2.4 Classificação Taxonômica dos Trabalhos	55
2.2.5 Bibliometria.....	57
3 3 MODELAGEM MATEMÁTICA PARA O PROBLEMA DE ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS COM MOVIMENTOS CONTÍNUOS.....	60
3.1 MODELAGEM MATEMÁTICA PRÉVIA.....	60
3.2 CARACTERÍSTICAS NÃO ABORDADAS.....	63
3.3 PROPOSTA DE MELHORIA DO MODELO MATEMÁTICO	65
3.4 IMPLEMENTAÇÃO E RESULTADOS PARCIAIS DO MODELO MELHORADO	68
4 O ALGORITMO DE OTIMIZAÇÃO DA COLÔNIA DE FORMIGAS	75

4.1 A REPRESENTAÇÃO DO PROBLEMA.....	76
4.2 O MOVIMENTO DAS FORMIGAS	78
4.3 A FASE DE MELHORIA.....	81
4.4 O ALGORITMO COMPLETO	82
4.5 DEFINIÇÃO DE PARÂMETROS PARA O ALGORITMO ACO	84
4.6 RESULTADOS ALCANÇADOS	85
4.7 CONSIDERAÇÕES SOBRE O ALGORITMO ACO APLICADO AO PCARP-CM90	
5 ITERATIVE RESTRICTED SPACE SEARCH APLICADO AO PROBLEMA DE	
ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS COM MOVIMENTOS	
CONTÍNUOS.....	91
5.1 ITERATIVE RESTRICTED SPACE SEARCH – IRSS.....	91
5.1.1 Detalhamento das fases.....	94
5.1.1.1 Fase de restrição.....	94
5.1.1.2 Processo de busca e a estrutura de árvore binária	97
5.2 APLICAÇÃO DO IRSS E RESULTADOS.....	99
5.2.1 Uso do IRSS no PCARP-CM.....	100
5.2.2 A fase de restrição	101
5.2.3 A fase de busca.....	106
5.2.4 Estratégia IRSS Exata.....	106
5.2.5 Estratégia IRSS Heurística.....	113
6 RESULTADOS GERAIS	120
7 CONSIDERAÇÕES FINAIS	125
REFERÊNCIAS.....	128

1 INTRODUÇÃO

Os Problemas de Roteamento em Arcos (*Arc Routing Problem* ARP) têm grande aplicabilidade e são muito estudados. Eles consistem na definição de rotas que têm como objetivo cobrir ruas, rodovias, ferrovias e vias em geral, as quais, quando modeladas, são transformadas em arcos de um grafo. A tomada de decisões neste tipo de problema pode ser auxiliada com a criação de modelos matemáticos e, em geral, quanto mais próximo o modelo à realidade, mais restrições devem ser impostas para que ele possa ser representativo. Desta forma, os modelos vêm se tornando cada vez mais complexos, retratando melhor diversas realidades.

Em 1981, Golden e Wong (GOLDEN; WONG, 1981) propuseram uma condição importante relativa aos ARP, a de que os veículos passam a ter capacidade limitada executando tarefas ao longo da sua rota. Isto é, pode ser que em dado momento o veículo atinja sua capacidade máxima e não dê continuidade à execução de suas tarefas. A partir disso, foi proposto o Problema de Roteamento Capacitado em Arcos (*Capacitated Arc Routing Problem* - CARP), um dos problemas mais estudados e abordados da classe de ARP.

O CARP, por sua definição, tem como objetivo criar rotas considerando veículos com capacidade limitada, os quais devem fazer coletas ou entregas em uma rede (grafo) de vias conectadas. As rotas que podem ser definidas por meio do CARP fazem parte de um processo de decisão habitualmente de nível operacional. São decisões de curto prazo envolvendo todas as operações que devem ser feitas para um pequeno período. Porém, em diversas situações é necessário fazer combinações dessas decisões considerando um horizonte de tempo maior, o que implica outro tipo de problema. Por exemplo, na coleta de lixo dificilmente necessita-se fazer coletas em todas as ruas em um único dia. Quando o problema é modelado para um horizonte de tempo maior que um dia, procurando combinar rotas que acarretem em um menor custo, tem-se o que é chamado de Problema de Roteamento Periódico e Capacitado em Arcos Capacitados, *Periodic Capacitated Arc Routing Problem* (PCARP).

O PCARP é uma generalização do CARP para múltiplos períodos, no qual tanto decisões táticas como operacionais podem ser tomadas. Vale ressaltar que “um problema periódico é caracterizado por diferentes frequências de serviços dependendo das necessidades de seus consumidores e o espaçamento de serviço

entre visitas” (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a, p. 125, tradução nossa).

Aplicações do PCARP são encontradas, por exemplo, nas seguintes situações: coleta de lixo (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005), remoção de neve em países com inverno rigoroso, aplicação de herbicidas em ferrovias, entrega de correspondências, inspeção de linhas elétricas, monitoramento, inspeção e manutenção de rodovias (MARZOLF; TRÉPANIER; LANGEVIN, 2006; MONROY; AMAYA; LANGEVIN, 2013) e ferrovias (BATISTA, 2014), irrigação de estradas não pavimentadas em minas ao céu aberto (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a, 2014b, 2016), inspeção de encanamentos e irrigação de árvores na beira de estradas (HUANG; LIN, 2014).

O período é o grande diferencial do PCARP, pois ir além do nível operacional nas decisões é uma contribuição valiosa para o desenvolvimento científico e empresarial. Ao tratar de múltiplos períodos necessita-se definir a fração temporal que dimensiona cada período e como o modelo matemático a utilizará. Na literatura usualmente define-se que o período considerado está relacionado com dias e estipular quantas rotas forem necessárias para cada dia respeitando a capacidade dos carros. Recentes abordagens têm tratado os períodos de forma diferenciada (BATISTA; SCARPIN, 2015; RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014b, a, 2016). Nestas, o tempo final de algum movimento é o tempo inicial do próximo movimento, respeitando o fluxo das continuidade de rotas. Quando utilizada essa última abordagem, uma solução inteira pode ser afetada ao modificar-se um único movimento. Esta proposta difere das abordagens que visam montar rotas para dias específicos, nas quais não importa o momento em que o arco é atendido, seja no começo ou ao final do dia.

Este trabalho foca-se na definição do Problema de Roteamento Periódico e Capacitado em Arcos com Movimentos Contínuos, Periodic Capacitated Arc Routing Problem with Continuous Moves (PCARP-CM). Em alguns problemas de roteamento, principalmente quando se trata de problemas marítimos, os veículos envolvidos fazem movimentos que são chamados de contínuos. Esses movimentos são denominados dessa forma porque não têm a necessidade de ter um depósito vinculado a sua rota. No caso marítimo, os navios fazem rotas continuamente entre os portos, o destino final de cada rota passa a ser o ponto inicial de uma nova. Isso acontece também em alguns problemas específicos como pick-up and delivery

(coleta e entrega) e no truckload (carregamento de caminhões). Neste caso, por exemplo, caminhões são carregados, transportam suas cargas até um destino e, lá chegando, são descarregados, buscam por outra carga próxima e seguem viagem indefinidamente, repetindo-se sempre o mesmo processo.

Observa-se que os movimentos contínuos trazem consigo uma complexidade adicional (SAVELSBERGH; SONG, 2007) fugindo do tradicional alinhamento entre clientes e fornecedores, em que um fornecedor acaba sempre fazendo rotas para entregar aos mesmos clientes seus produtos. Outro exemplo de movimento contínuo ocorre na produção e distribuição de argônio (SAVELSBERGH; SONG, 2008), o qual é resultado da produção de oxigênio. As demandas por argônio nem sempre são compatíveis e a sua falta pode ser iminente visto que ele só é produzido se existe a produção de oxigênio. Para suprir a necessidade de argônio, muitos clientes são servidos por sleeper teams que ficam dias consecutivos nas ruas pegando argônio de diferentes fornecedores ao longo de seu caminho e fazendo distribuição. Sendo assim, os sleeper teams movem-se continuamente e cobrem uma grande área geográfica para garantir que consumidores de argônio possam receber seu produto. De modo que os clientes consomem argônio de diferentes fornecedores.

O PCARP-CM aqui abordado é referente ao problema aplicado na dissertação de Batista (2014), cujo trabalho foi tratado como PCARP sem receber uma classificação específica, mesmo contendo características diferentes das definições gerais. Esse nome segue a tendência de artigos recentes que têm dado nomes específicos às outras variações de PCARP, como: 1) com serviços irregulares (MONROY; AMAYA; LANGEVIN, 2013) e 2) com restrições de estoque (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014b).

O contexto do trabalho aqui proposto está relacionado à inspeção e manutenção de ferrovias, sendo uma continuidade do que foi proposto em Batista (2014). Veículos homogêneos e com capacidade limitada são responsáveis pela inspeção dos trilhos para garantir o fluxo nas vias. Essa capacidade é associada ao deslocamento diário que os carros conseguem fazer entre dois pontos, os quais são chamados de trechos. Cada trecho deve ser periodicamente avaliado a fim de uma rápida detecção de problemas que poderiam acarretar em acidentes. A periodicidade de cada trecho pode não ser respeitada, porém, quando isso ocorre, um custo de atraso, penalidade, deve ser levado em conta. Este problema tem

também uma característica cíclica, posto que as rotas determinadas para um horizonte de planejamento devem repetir-se indefinidamente.

Como a malha ferroviária, em um país como o Brasil, é extensa, os veículos devem fazer diariamente um grande percurso, tornando-se inviável seu retorno até um depósito. Logo, a rota deve ser contínua, caracterizando o problema como PCARP-CM. Desta forma, a periodicidade de cada trecho é dada segundo um intervalo de tempo máximo em que cada trecho deve ser analisado; as avaliações devem ocorrer dentro de um horizonte de tempo/planejamento limitado, o qual deve ser no mínimo igual à maior periodicidade; e a solução encontrada deve formar ciclos, pois o trabalho de monitoramento segue em regime permanente.

Será aqui proposta uma melhoria no modelo matemático de Batista (2014), considerando-se os fatores expostos anteriormente. No modelo prévio, os carros deveriam iniciar e terminar suas rotas no mesmo ponto depois que decorrido o horizonte de tempo, garantindo a adequação à característica cíclica do problema. Quando considerada a possibilidade de troca entre pontos finais e iniciais entre os carros, situação proposta pelo PCARP-CM, a validação da ciclicidade é feita garantindo que a quantidade de veículos que deixa um ponto no primeiro dia seja igual à quantidade de veículos que retornam a esse ponto no último dia, independentemente do carro, de modo que a solução segue cumprindo a demanda cíclica do problema.

Dentre os trabalhos até então publicados envolvendo PCARP, nenhum trouxe uma proposta taxonômica que classificasse uma ampla gama de características já estudadas. Simples classificações já foram propostas, mas levaram em conta poucos atributos do problema. No presente trabalho será feita uma análise cuidadosa de cada trabalho, o que culmina em uma proposta taxonômica que considera características físicas, foco prático ou teórico e métodos de resolução.

Resolver problemas aplicados, como o descrito, é uma tarefa complexa tanto para *solvers* comerciais quanto para métodos heurísticos. No PCARP, algoritmos exatos geralmente provam soluções ótimas para instâncias limitadas e pequenas (CHU; LABADI; PRINS, 2004c; MONROY; AMAYA; LANGEVIN, 2013). Existem muitas pesquisas com foco no desenvolvimento de heurísticas que permitam alcançar soluções de alta qualidade (CHEN; HAO, 2018), entre elas pode-se citar a aplicação de algoritmos heurísticos como Scatter Search (CHU; LABADI; PRINS, 2006); Algoritmos Meméticos (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005; MEI

et al., 2011; ZHANG *et al.*, 2016); Adaptive Large Neighborhood Search (ALNS) (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a); e Otimização de Colônia de Formigas, Ant Colony Optimization (ACO), (HUANG; LIN, 2014; KANSOU; YASSINE, 2009).

Entre os algoritmos citados, o ACO é um dos que apresenta bons resultados para problemas de roteamento dinâmico, nos quais os custos podem ser modificados após algum movimento (DORIGO; STÜTZLE, 2010). O PCARP-CM tem semelhanças com estes problemas, pois tem demandas atualizadas em cada período que podem interferir na função objetivo. Com base nisso, será introduzido um novo Algoritmo ACO que alcança resultados competitivos quando comparado ao resultado obtido por um *solver* comercial. A nova abordagem da ACO combina conceitos de Busca Tabu Granular, Granular Tabu Search (GTS) e busca local (TOTH; VIGO, 2003).

Diante do exposto acima é possível verificar que ainda existe campo para pesquisa quando tratado do PCARM-CM. O presente trabalho será desenvolvido tomando como base os objetivos descritos a seguir.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Formalizar o Problema de Roteamento Periódico e Capacitado em Arcos com Movimentos Contínuos classificando-o segundo suas características e resolvendo-o de forma heurística e exata.

1.1.2 Objetivos específicos

Os objetivos específicos são os seguintes:

- Revisar alguns dos principais problemas abordados na literatura acerca de roteamento em arcos;

- Propor uma classificação taxonômica para o PCARP que englobe características físicas de cada problema assim como seu foco e abordagem resolutiva;
- Fazer uma análise bibliométrica acerca do tema;
- Propor uma melhoria para o modelo matemático desenvolvido em Batista (2014) que aborda movimentos contínuos;
- Propor e aplicar um novo algoritmo de Otimização de Colônia de Formigas no problema que usa conceitos de Busca Tabu Granular e Busca Local;
- Utilizar a solução do ACO como solução inicial para o *solver* resolver o problema de maneira exata;
- Aplicar uma técnica de busca iterativa em subespaços do espaço solução do problema chamado Busca Iterativa em Espaço Restrito, *Iterative Restricted Space Search* (IRSS), por meio de uma abordagem exata e uma heurística;
- Analisar e comparar os resultados obtidos.

1.2 CONTRIBUIÇÕES E LIMITAÇÕES

O presente trabalho apresenta três contribuições importantes. A primeira é a classificação dos artigos que tratam do PCARP. Com a classificação é possível verificar quais as abordagens mais utilizadas como um todo, as variações que podem compor um problema e as lacunas encontradas, ou seja, espaços que podem conter problemas ainda não estudados. O problema começou a ser tratado apenas em 2002 e, ainda que a quantidade de publicações não tenha a mesma dimensão dos problemas clássicos de roteirização em nós, é possível perceber que com o decorrer do tempo vêm sendo feitas mais aplicações e criadas variantes.

O fato de ser um tema recente acarreta em uma certa dificuldade dos próprios autores em classificar suas variações, logo, alguns problemas semelhantes são tratados com nomes diferentes, ou se posicionam mal quanto às nomenclaturas mais utilizadas. Por exemplo, Problema de Roteamento Periódico em Arcos com Pontos de Refil (HUANG; LIN, 2014), ou ainda, Problema Periódico do Carteiro

Rural (GHIANI *et al.*, 2005) são trabalhos em que os autores poderiam atribuir nomes que os classificassem como um PCARP. A proposta de uma taxonomia, contribuição desse trabalho, permitirá que novos trabalhos sejam bem classificados, auxiliando inclusive em como nomeá-los.

A segunda contribuição é o aperfeiçoamento do modelo de Batista (2014). Uma característica importante é adicionada ao permitir que carros troquem seus pontos iniciais ao final de um ciclo. Além disso, é importante a classificação desse problema como com Movimentos Contínuos, pois tem sido uma tendência nos trabalhos recentes.

A última contribuição é quanto à resolução do problema, como o PCARP-CM ainda não foi explorado por outros autores, nenhum algoritmo até então foi proposto para criar soluções que considerem as peculiaridades do modelo. Tem-se a aplicação de um novo algoritmo ACO que combina conceitos de busca tabu para inviabilizar movimentos que não agregam valor a função objetivo e a aplicação de operadores de melhoria local. O ACO usa uma série de informações em seu processo e os dados disponíveis da resolução são base para criação de um método híbrido. O método híbrido, chamado Iterative Restricted Space Search (IRSS), usa uma ferramenta exata para buscar soluções em uma região limitada do espaço solução de um problema. Cada espaço é restringido utilizando as informações do ACO e, de maneira iterativa, esses espaços restritos são avaliados um a um.

1.3 ESTRUTURA DO TRABALHO

O capítulo 2 traz um levantamento geral dos trabalhos que envolvem roteamento em arcos, dando ênfase ao PCARP e sendo apresentada uma proposta taxonômica de acordo com os critérios previamente expostos. Ao final do capítulo também pode ser encontrada uma breve bibliometria.

No capítulo 3 o modelo matemático desenvolvido em Batista (2014) é apresentado e discutido. As discussões visam permitir o aperfeiçoamento da modelagem, podendo alcançar uma quantidade maior de soluções e de qualidade no mínimo igual ao modelo anterior. Ao final do capítulo encontram-se novas instâncias e os resultados dessa nova modelagem.

O capítulo 4 apresenta uma alternativa a abordagem exata, uma nova heurística de Otimização de Colônia de Formigas (ACO) para o PCARP-CM. O

algoritmo é novo por usar conceitos de Busca Tabu Granular de forma a evitar com que formigas façam movimentos desinteressantes e ainda aplicar uma melhoria de busca local. Os resultados mostram a boa performance do algoritmo em casos críticos em que a ocupação mínima dos carros é alta. Uma primeira hibridização é apresentada ao usar a solução do ACO como ponto de partida para *solver*.

Já no capítulo 5 é apresentada o algoritmo de Busca Iterativa em Espaço Restrito (IRSS). A ACO conduzirá a limitação de regiões interessantes do espaço solução formando os chamados espaços restritos que serão explorados por um algoritmo exato, nesse caso, o *solver*. A forma com que os espaços são explorados pode trazer uma característica exata ou heurística ao método que alcançou bons resultados para o PCARP-CM.

O capítulo 6 traz um comparativo entre todas as técnicas utilizadas, suas performances e resultados alcançados. Finalizando, o capítulo 7 apresenta as considerações finais dessa tese.

2 PCARP: REVISÃO DE LITERATURA E PROPOSTA TAXONÔMICA

Os problemas de roteamento são amplamente estudados na literatura, podendo basicamente ser divididos em duas classes: roteamento em nós e roteamento em arcos. A natureza do serviço ou da demanda dita a qual dessas classes o problema pertence. O roteamento em arcos ocorre quando existe uma alta densidade de pontos em uma linha, sendo convertido de um problema de roteamento em nós para arcos, ou quando o interesse é atravessar arcos (ASSAD; GOLDEN, 1995).

Esse capítulo focará nos problemas de roteamento em arcos e sua evolução até chegar ao PCARP. Alguns trabalhos de roteamento de veículos associados ao roteamento em nós serão citados para mostrar algumas das mais variadas características que os problemas podem apresentar.

2.1 O ROTEAMENTO EM ARCOS

Um apanhado dos problemas de roteamento em arcos estudados entre 2000 e 2010 pode ser encontrada em Corberán e Prins (2010). A origem desses problemas está ligada ao Problema das Pontes de Königsberg, os quais tentam encontrar uma ou mais rotas que cubram todas (ou parcialmente) as ligações (arcos ou arestas) de um grafo, satisfazendo algumas restrições com o menor custo possível.

Muitos fatores podem influenciar os problemas de roteamento, aumentando sua complexidade. Entre os fatores podem-se encontrar: número de domicílios, tempo para servir um determinado nó ou arco, tamanho da frota, natureza da demanda, localização da demanda, restrições na capacidade dos veículos, natureza do grafo, entre outros (KONOWALENKO, 2012).

O principal objetivo do Problema de Roteamento em Arcos (PRA), em inglês *Arc Routing Problems*, é determinar o menor custo para atravessar um conjunto de arcos de um grafo, com ou sem restrições (EISELT; GENDREAU; LAPORTE, 1995a). É possível fazer a conversão de problemas de roteamento em arcos para roteamento em nós e vice-versa. Essa conversão é custosa pois em geral aumenta o número de variáveis, não sendo utilizada com frequência (ASSAD; GOLDEN, 1995).

Alguns dos problemas de roteamento em arcos já estão bem consolidados na literatura e sofrem variações que visam deixar o comportamento dos modelos cada vez mais próximos à realidade. É o caso do Problema do Carteiro Chinês, o Problema do Carteiro Rural e o Problema de Roteamento Capacitado em Arcos.

O *Chinese Postman Problem* (CPP), ou Problema do Carteiro Chinês, é ambientado no contexto de um carteiro que deve partir de um depósito e entregar suas cartas ao longo de um conjunto de ruas para, em seguida, voltar ao depósito (TAUBE, 1992). Ele serviu de inspiração para muitas aplicações em problemas reais nas últimas décadas (CORBERÁN; MOTA; SANCHIS, 2006). Em Hsiao-Fan e Yu-Pin (2002) o CPP é classificado segundo seu grafo de três modos distintos: não direcionado, direcionado e misto; modelagens matemáticas para os referidos problemas estão também disponíveis em seu trabalho. O CPP misto, assim como uma variação chamada de Problema do Carteiro Chinês com Vento, está entre os problemas mais estudados na primeira década dos anos 2000 (CORBERÁN; PRINS, 2010). Variantes do CPP encontradas na literatura são: CPP Hierárquico (GHIANI; IMPROTA, 2000), CPP com Benefício Máximo (MALANDRAKI; DASKIN, 1993) e CPP com Janelas de Tempo (HSIAO-FAN; YU-PIN, 2002).

O Problema do Carteiro Rural, *Rural Postman Problem* (RPP), se assemelha ao CPP. A diferença entre os problemas se dá pelo fato de que no RPP não é necessário atender todos os arcos do grafo. Nele existe um subconjunto de arcos chamado de arcos requeridos, os quais devem ser atravessados ou servidos. Cada arco requerido deve ser atravessado uma única vez e um caminho de custo mínimo deve ser encontrado (FERNÁNDEZ *et al.*, 2003). Também é uma classe de problemas que tem recebido grande atenção dos pesquisadores (CORBERÁN; PRINS, 2010).

A maior parte dos problemas reais eram usualmente modelados como RPP, pois poucos problemas requerem que todos os arcos sejam servidos (EISELT; GENDREAU; LAPORTE, 1995b). Outras aplicações podem envolver uma quantidade maior de veículos e limitantes de capacidade, distância ou tempo viajado. Problemas como o de limpeza de ruas, remoção de neve, coleta de lixo, entrega de cartas, rotas de ônibus escolares e leituristas (os responsáveis por fazer a leitura do consumo de água e/ou energia elétrica) foram tratados como RPP. O Problema do Carteiro Rural Periódico (GHIANI *et al.*, 2005) é uma variante que vem ao encontro do desenvolvimento desse trabalho, visto que uma rota deve ser obtida

com custo mínimo, é utilizada para múltiplos períodos e que cada arco deve ser visitado um número de vezes previamente determinado no decorrer desse longo período.

Normalmente os veículos tem uma determinada capacidade que está associada a uma carga máxima, a uma máxima capacidade de trabalho ou a uma restrição de distância máxima a ser percorrida (CORBERÁN; PRINS, 2010). Ao tentar aproximar os modelos de Pesquisa Operacional à realidade, uma questão é proposta: “o que aconteceria se introduzisse uma coleta ou entrega ao atravessar um arco?” (DROR, 2000). A resposta para essa questão foi a criação do Problema de Roteamento Capacitado em Arcos, *Capacitated Arc Routing Problem* (CARP), que combina essas situações.

O CARP foi introduzido em Golden e Wong (1981), um problema rico em teoria e aplicabilidade que era negligenciado até então. Uma demanda é associada a cada arco de um grafo e veículos de uma frota com capacidade W devem atravessar esses arcos coletando ou entregando as determinadas demandas sem exceder a capacidade. Modelos matemáticos podem ser encontrados em Dror (2000) e Golden e Wong (1981). Em 1995, o CARP foi considerado o problema mais importante na área de roteamento em arcos (EISELT; GENDREAU; LAPORTE, 1995b). Ele se configura como um problema de complexidade NP-Hard (GOLDEN; WONG, 1981) e muitas abordagens de resolução já foram feitas, como algoritmos baseado em Busca Tabu (HERTZ; LAPORTE; MITTAZ, 2000), *Guided Local Search* (BEULLENS *et al.*, 2003), planos de corte (BELENGUER, JM; BENAVENT, 2003) e Algoritmos Genéticos em um problema transformado em Problema de Roteamento de Veículos Capacitado (LONGO; DE ARAGÃO; UCHOA, 2006).

Uma quantidade de 25 problemas teste, instâncias, de uma variação específica de CARP, Problema do Carteiro Chinês Capacitado, foram propostas em Golde, Dearmon e Baker (1983) para realizar experimentos computacionais. Essas instâncias são referências clássicas, porém apenas 23 das 25 instâncias são usadas recorrentemente e adaptadas em diversos trabalhos, porque 2 instâncias (8 e 9) contêm inconsistências (CORBERÁN; PRINS, 2010).

Diversas aplicações são exploradas e muitas variações no problema podem ser encontradas, são elas: custos dependentes do tempo (TAGMOUTI; GENDREAU; POTVIN, 2007) determinando o momento exato de servir cada arco; conversões proibidas (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002b); CARP com demanda

“morta”, CARP *with Deadheading Demand* (BARTOLINI; CORDEAU; LAPORTE, 2013); CARP quando um arco não precisa ser servido por apenas um veículo, *Split-Delivery* CARP (BELENGUER *et al.*, 2010); Problema de Roteamento em Arcos e *Scheduling* com transbordo (ROSA *et al.*, 2002); CARP contendo Rotas Abertas ou Fechadas (FUNG; LIU; JIANG, 2013; USBERTI, 2012; USBERTI; FRANÇA; FRANÇA, 2011); com demanda estocásticas (LAPORTE; MUSMANNO; VOCATURO, 2010); e com pontos de retil (AMAYA; LANGEVIN; TRÉPANIER, 2007).

Diversas alternativas são adotadas na resolução desses problemas: transformação em um problema de roteamento em nós e geração de colunas (TAGMOUTI; GENDREAU; POTVIN, 2007); planos de corte e algoritmos genéticos (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002b); *Multistart Evolutionary Local Search* (BELENGUER *et al.*, 2010); Busca Tabu (ROSA *et al.*, 2002); GRASP (USBERTI, 2012; USBERTI; FRANÇA; FRANÇA, 2011); entre outras.

O CARP básico da literatura envolve decisões de um único período em um grafo não direcionado, ou seja, cada arco de uma rede pode ser atravessado em qualquer direção (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a). A realidade de muitos problemas acaba, muitas vezes, não sendo bem representada pelo CARP devido à sua limitação temporal, de modo que uma das mais importantes variações ocorre quando é exigido um planejamento de rotas dentro de um horizonte de tempo prolongado.

2.2 O PROBLEMA DE ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS

Para decisões que são tomadas diariamente, o CARP é muito utilizado e eficaz. Quando é necessário abordar problemas que vão além do nível operacional, sendo ele o nível tático ou estratégico o CARP deixa a desejar. Trabalhar com um horizonte de tempo maior implica em problemas mais complexos de serem resolvidos, dando origem ao Problema de Roteamento Periódico e Capacitado em Arcos, *Periodic Capacitated Arc Routing Problem* (PCARP). Nas primeiras vezes que o PCARP foi abordado (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, 2002c), ele foi chamado de *Arc Routing Planning Problem*, problema de planejamento em roteamento em arcos, recebendo esse nome por fazer uma correspondência com o

problema de planejamento da produção em que uma decisão tática deve ser tomada.

A importância do período no roteamento de veículos já vem sendo abordada há muito tempo, pois isso permite a redução de custos de distribuição ao longo dos dias (CHRISTOFIDES; BEASLEY, 1984). Um trabalho pioneiro relacionado ao assunto pode ser encontrado em Eglese (1994), no qual é proposto um roteamento com características de um PCARP. Características como frequência, capacidade e roteamento são abordadas, porém sua resolução permanece apenas no nível operacional. O problema proposto por Eglese (1994) estava relacionado a um contexto de remoção de neve nas vias dos países com inverno rigoroso, situação na qual algumas vias precisavam ser tratadas com sal a cada duas horas e outras a cada quatro horas.

Uma das definições que mais influenciaram o uso do PCARP é a de Lacomme, Prins e Randane-Chérif (2005):

“O problema de roteamento de veículos capacitado e periódico em arcos consiste aqui na seleção para cada rua de um número dias de tratamento igual a sua frequência (decisão tática) e então formar uma rota em cada dia de forma que as ruas designadas para esse determinado dia sejam atendidas (decisão operacional). Obviamente, o custo total no final do período depende da combinação das decisões de designação e roteamento.” (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005, p. 537, tradução nossa)

A definição foi baseada na aplicação do problema da coleta de lixo, diante do fato de que as empresas responsáveis pelas coletas reúnem várias estatísticas sobre a produção diária de lixo em cada região. Os grandes centros acabam necessitando atendimentos mais frequentes do que regiões mais afastadas, logo um planejamento baseado em múltiplos períodos é essencial.

O PCARP pode ser visto como uma extensão natural do CARP, na qual ao invés do problema ser tratado para um único dia, ele é então expandido para múltiplos períodos (MONROY; AMAYA; LANGEVIN, 2013). Vale ressaltar que aproximações por meio do simples modelo do CARP são dificilmente aplicadas (MEI *et al.*, 2011), sendo assim o PCARP pode representar melhor muitas realidades. A resolução do PCARP implica conciliar as decisões de nível tático e operacional simultaneamente levando em conta todo o horizonte de tempo. A realidade é que o problema pode conter algumas complicações como, por exemplo, demandas

flutuantes ao longo do horizonte de tempo ou espaçadas em intervalos de tempo (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005).

O PCARP vai mais longe quando analisada a proposta de Riquelme-Rodríguez, Gamache e Langevin (2016) em que o problema de localização de facilidades é incorporado ao contexto do PCARP. Uma decisão estratégica deve ser combinada com a operacional, pois a definição da localização dos depósitos é uma decisão de longo prazo. Alguns outros autores ressaltam que o objetivo primário do PCARP deve ser o de reduzir o número de veículos e depois minimizar o custo total (MEI *et al.*, 2011). Isso torna o PCARP um problema de otimização hierárquica, diferentemente dos problemas com objetivo único.

Aplicações diferenciadas do PCARP vêm sendo publicadas recentemente. Na tentativa de englobar todas suas variantes, o PCARP pode ser definido como: Problema de roteamento em arcos envolvendo um horizonte de tempo composto por mais de um período, sendo que estes períodos podem ser formados por qualquer fração temporal desde que discreta. Um ou mais veículos ao fazerem suas rotas devem executar da melhor maneira possível as tarefas que estão associadas a cada arco sem exceder sua capacidade e respeitando as frequências exigidas.

A abordagem padrão do PCARP é possuir custos associados ao atravessamento dos arcos e outros custos para servi-los. São custos não negativos que podem até mesmo ser iguais pela dificuldade em estimá-los, ou simplesmente por serem iguais. Um arco pode ser atravessado quantas vezes forem necessárias, porém o atendimento das tarefas deve ocorrer em momentos planejados. É assumido também que as demandas individuais nunca excedem a capacidade do carro, a exemplo do CARP, para evitar atendimentos germinados (CHU; LABADI; PRINS, 2006).

O PCARP está restrito ao que pode ser transportado por um veículo (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014b), isto é, pode-se entender como o aproveitamento máximo do esforço/capacidade deste. É um problema considerado NP-Hard por englobar o CARP, além disso, por ser periódico, pode ter um espaço de solução muito maior que o CARP. É necessário resolver um problema de alocação ao definir os dias em que os arcos devem ser servidos (ZHANG *et al.*, 2016).

O problema equivalente ao PCARP no roteamento em nós seria o Problema de Roteamento Periódico de Veículos (PRPV). Diversas heurísticas são usadas para

a resolução desse problema, sendo um exemplo o algoritmo híbrido de GRASP com Busca Tabu (TORTELLY JR.; OCHI, 2003). Variações do PRPV também são abordados, como: a existência de múltiplos depósitos (VIDAL *et al.*, 2012); a frequência de visitas sendo definida pelo modelo (FRANCIS; SMILOWITZ, 2006); entre outros.

Na próxima seção será possível encontrar um detalhamento dos trabalhos quanto a suas aplicações e abordagens resolutivas. Cada problema abordado na literatura como PCARP pode ter diferentes características físicas ou estruturais que podem interferir na resolução. As principais características estão listadas e explicadas em uma proposta taxonômica após o detalhamento dos trabalhos.

2.2.1 Contribuições dos trabalhos científicos acerca do PCARP

Esta seção tem por objetivo discutir as principais contribuições científicas a respeito do PCARP, um pouco de sua história, mostrar os contextos reais em que alguns dos trabalhos se aplicaram e as abordagens de resolução adotadas.

Sem dúvida, o PCARP teve seu crescimento e desenvolvimento baseado em aplicações. Os primeiros trabalhos publicados foram desenvolvidos inspirados em problemas de remoção de neve, inspeção de linhas elétricas e, principalmente, na coleta de lixo. O primeiro trabalho encontrado sobre PCARP é o de Lacomme, Prins e Ramdane-Chérif (2002c) que foi publicado em uma conferência em abril de 2002, em seguida, em julho, mais detalhes foram exibidos em outra conferência (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a), ambos trabalhos culminam no artigo intitulado “*Evolutionary algorithms for periodic arc routing problems*” (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005).

Prins, um dos autores desses artigos esteve trabalhando com esse assunto com outros autores e teve quatro publicações em eventos nas datas: outubro de 2002 (CHU; LABADI; PRINS, 2002), maio de 2003 (CHU; LABADI; PRINS, 2003), abril de 2004 (CHU; LABADI; PRINS, 2004a) e julho de 2004 (CHU; LABADI; PRINS, 2004b), dentre estes o único artigo acessível é o de Chu, Labadi e Prins (2003). Ao serem analisados os títulos destes trabalhos é possível supor que essas apresentações em congressos resultaram nos artigos destes autores publicados nos anos de 2004, 2005 e 2006 (CHU; LABADI; PRINS, 2004c, 2005, 2006).

Os trabalhos pioneiros (CHU; LABADI; PRINS, 2004c, 2005, 2006, LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, 2002c, 2005), apesar de terem como principal objetivo a apresentação do PCARP em uma forma geral, também mostram que aplicações podem ser feitas ao problema de inspeção de linhas elétricas, despejamento de herbicidas em ferrovias, remoção de neve em rodovias e um destaque especial é dado para o problema da coleta de lixo. Isso talvez, pela fácil associação de alguns conceitos: ruas são associadas a arcos, cruzamentos a nós, frequências ao número de coletas na semana, demandas às quantidades de lixo e capacidade dos veículos a máxima quantidade de lixo que cabe em um caminhão. Os habitantes de uma cidade produzem lixo continuamente e o armazenam em sacos plásticos ou containers. A coleta deve ser bem planejada, pois algumas regiões têm produção maior que outras e necessitam mais atendimentos. Também se verifica nestes trabalhos pioneiros a preocupação em estar usando uma quantidade mínima de veículos.

Para validar os testes iniciais, tanto Lacomme, Prins e Ramdane-Chérif (2005) como Chu, Labadi e Prins (2004c, 2005, 2006), utilizaram o mesmo processo para a adaptação de 23 instâncias advindas de Golden, Dearmon e Baker (1983). Apesar de existir um pequeno detalhe na descrição do processo que poderia diferenciar o processo de adaptação, quando as estruturas dos dados são mostradas é possível observar que são iguais. Essas instâncias são as mais importantes da literatura, chamadas de *pgdb* e usadas em Kansou e Yassine (2009), Mei *et al.* (2011) e Zhang *et al.* (2016). Outras instâncias foram adaptadas do trabalho de Belenguer e Benavent (2003) em Chu, Labadi e Prins (2006) e em Lacomme, Prins e Ramdane-Chérif (2005). No trabalho de 2006 foram adaptadas 34 instâncias, recebendo o nome de *pval*, e em 2005, 10 novas instâncias surgiram e foram chamadas de *pegl*, sendo as maiores baseadas no condado de Lancashire (Inglaterra).

Um algoritmo evolucionário, Algoritmo Memético (AM), é proposto em Lacomme, Prins e Ramdane-Chérif (2005), mas a falta de lower bounds não permitia uma completa avaliação do algoritmo. Desta forma, ele foi comparado com uma Heurística de Melhor Inserção (HMI) e outras três diferentes aplicações que utilizaram o AM, sendo que o resultado do HMI sempre estava nas populações iniciais. O primeiro AM resolveu simultaneamente as decisões táticas e operacionais, recebendo o nome de Algoritmo Memético Periódico (AMP); o segundo aplica o AM

após as tarefas terem sido alocadas a dias por meio do HMI; e o terceiro faz uma pós otimização baseada nos resultados do AMP. Os resultados foram apresentados para as instâncias *pgdb* e *pegl*. Abordagens semelhantes relacionadas a Algoritmos Genéticos estão presentes em Lacomme, Prins e Ramdane-Chérif (2002a, 2002c).

Um primeiro modelo matemático utilizando combinação de dias é desenvolvido em Chu, Labadi e Prins (2005). O modelo é resolvido no *solver* Xpress, no qual foi possível alcançar soluções ótimas para problemas muito pequenos com 5 períodos e 10 arcos. Provavelmente esse modelo deve ter sido apresentado anteriormente em uma sessão oral (CHU; LABADI; PRINS, 2002) como referenciado em Chu, Labadi e Prins (2006). A abordagem de resolução adotada foi a criação de 3 heurísticas: dois métodos de inserção e um algoritmo de duas fases (agrupamento e roteirização). As heurísticas consistiam na construção de listas de viagens para cada dia do horizonte de tempo.

O trabalho de Chu, Labadi e Prins (2006) apresenta o algoritmo *Scatter Search*, uma abordagem que usa combinação de dias. *Scatter Search* é um algoritmo evolutivo que opera com uma pequena quantidade de soluções, combinando-as para gerar novas soluções, sendo que dentro do processo existe uma probabilidade de ocorrer uma busca local. *Lower bounds* são apresentados, porém não representam bem o problema como um todo, para efeitos comparativos é usada uma heurística de melhor inserção. Além disso, o desempenho do algoritmo é avaliado por meio de aplicações em instâncias CARP, conseguindo bons resultados. Mais detalhes sobre os *lower bounds* podem ser encontrados também em Chu, Labadi e Prins (2003), que é dedicado a esta temática.

O trabalho de Chu, Labadi e Prins (2004c), apesar de publicado antes dos artigos dos mesmos autores de 2005 e 2006 (CHU; LABADI; PRINS, 2005, 2006), foi feito após eles. Neste trabalho (CHU; LABADI; PRINS, 2004c) é proposto o primeiro modelo matemático não considerando um subconjunto de dias para designação. Também é dado destaque para os *lower bounds* encontrados e suas provas, as quais consistiam em transformar o PCARP em um CARP maior. Novamente o modelo matemático foi resolvido com o software Xpress conseguindo soluções ótimas apenas para pequenas instâncias. Com os *lower bounds* foi possível mensurar a qualidade da solução obtida com o algoritmo *Scatter Search* proposto em Chu, Labadi e Prins (2006).

Um outro trabalho pioneiro e basicamente teórico, é o de Ghiani *et al.* (2005), o qual, apesar de referenciado como 2005 tem um registro de 2003 e aborda o Problema do Carteiro Rural Periódico. Possíveis aplicações em coleta de lixo e limpeza de ruas são brevemente mencionadas. Neste problema, as ruas devem ser servidas uma certa quantidade de vezes igualmente espaçadas ao longo do horizonte de tempo utilizando combinações de dias. Os próprios autores relatam a pobreza e falta de organização da literatura acerca dos problemas de roteamento em arcos, o que pode tê-los impossibilitado de classificá-lo como um problema Periódico de Roteamento em Arcos, pois o trabalho também não leva em conta capacidade alguma, ou ainda, as capacidades seriam infinitas. Não é apresentado modelo matemático e uma heurística foi desenvolvida e aplicada a 80 instâncias, nas quais as soluções ótimas eram conhecidas a priori, para testar a eficiência do algoritmo proposto.

O PCARP com serviços irregulares, *Periodic Capacitated Arc Routing Problem with Irregular Services* (PCARP-IS) proposto em Monroy, Amaya e Langevin (2013) é usado no contexto de vigilância, monitoramento e até mesmo manutenção de rodovias. O contexto é o mesmo do trabalho de Marzolf, Trépanier e Langevin (2006), em que ao longo de um horizonte de tempo de duas semanas as rodovias devem ser monitoradas e, para isso, as vias são divididas hierarquicamente e algumas necessitam ter um monitoramento maior que outras. Quando se tratando de monitoramento, o objetivo de minimizar os custos com deslocamentos pode se tornar irrelevante, visto que quanto mais vezes ocorrer o monitoramento de algumas vias, mais garantida será a segurança.

O trabalho de Marzolf, Trépanier e Langevin (2006) utiliza apenas um carro, responsável pelo monitoramento e atendimento de ocorrências nas vias, estando disponíveis 3 turnos por dia e trabalhando com rotas pré-determinadas. Como o veículo tinha mais de uma função eles desenvolveram um ambiente onde a rota planejada deveria ser recalculada no caso de alguma ocorrência. Os resultados alcançados foram bons visto que algumas vias não eram monitoradas em virtude do atendimento a emergências (MARZOLF; TRÉPANIÉ; LANGEVIN, 2006). Por sua vez, o trabalho de Monroy, Amaya e Langevin (2013) deu ênfase em caracterizar os serviços irregulares e apresentar uma formulação matemática para o problema como um todo considerando capacidades permissíveis em cada rota.

O PCARP-IS (MONROY; AMAYA; LANGEVIN, 2013) apresenta uma modelagem matemática para o problema e alcança resultados para 5 grupos de instâncias, 3 criadas e outras 2 modificadas a partir de Benavent *et al.* (1992) e Golden e Wong (1981). Instâncias pequenas foram resolvidas e as demais foram abordadas por meio de uma heurística baseada em duas fases, primeiramente agrupamento e em seguida roteirização, *Cluster-First Route-Second*, usando sempre o *solver* Xpress-MP. O trabalho de Marzolf, Trépanier e Langevin (2006) estava no mesmo contexto. Nele foram usados dados reais e resolvidos mediante a aplicação de 3 métodos baseados em rotas pré-determinadas, eram eles: designação de rotas pré-determinadas, replanejamento e integração parcial de rotas. Dentro dos métodos foram usados modelos de designação e caminho mínimo resolvidos pelo Cplex 8.0.

Dentro do contexto de monitoramento também se encontra a inspeção e manutenção de ferrovias em Batista (2014). Trechos de ferrovias devem ser inspecionados segundo uma frequência baseada em intervalos de tempo. Como a movimentação para fazer inspeção é lenta e a malha ferroviária é extensa, os veículos envolvidos na aplicação não são vinculados a um depósito. Cada trecho está relacionado a um arco com uma respectiva periodicidade, que se não for atendida é associada a penalidades na função objetivo. Ao final do horizonte de tempo cada carro deve voltar ao ponto que iniciou seu serviço no primeiro dia. Um problema real representando uma malha ferroviária é apresentado, outros problemas testes foram gerados e as soluções obtidas foram encontradas por meio de métodos exatos utilizando o *solver* CPLEX 12.4. Uma tentativa de comparação entre formulações é feita usando a modelagem de serviços irregulares, porém os resultados da adaptação são piores que a do modelo apropriado (BATISTA; SCARPIN, 2014).

O PCARP envolvendo restrições de estoque, *Periodic Capacitated Arc-Routing Problem with Inventory Constraints* (PCARP-IC) é uma valiosa aplicação apresentada por Riquelme-Rodríguez, Gamache e Langevin (2014a, 2014b, 2016). Os autores afirmam que:

“Porque as demandas são localizadas nos arcos de uma rede e os arcos exigem serviços mais de uma vez dentro do horizonte de tempo, esse problema pertence ao domínio dos problemas de roteamento em arcos capacitado e periódico.” (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2016, p. 94, tradução nossa)

Essa problemática tem foco no problema de irrigação em minas ao céu aberto, reduzindo custos com transporte e atrasos. Uma frota de caminhões utiliza vias não pavimentadas que levantam pó quando são atravessadas, como as vias tem uma natureza temporária, soluções permanentes como pavimentação acarretam em um alto custo. O problema consiste na irrigação das vias, pois a umidade aumenta a retenção de partículas de pó, para evitar que muito pó fique espalhado pelo ar podendo afetar equipamentos e até mesmo a saúde humana.

A água despejada nas vias acaba evaporando, então se faz necessário despejar a água periodicamente. A periodicidade é influenciada pelas condições climáticas e, também, pela quantidade de veículos que passa em cada via. Os arcos se comportam como consumidores e uma quantidade suficiente de material deve ser entregue a ele, garantindo que sempre esteja em um nível desejado. As rodovias são classificadas de acordo com suas necessidades e os controles são feitos medindo níveis de umidade, neste âmbito as decisões de roteamento e estoque devem ser tomadas simultaneamente. A fração temporal não está restrita a dias nos quais rotas devem ser desenvolvidas, trabalhando-se com a ideia de continuidade do tempo: cada movimento é terminado em um instante no futuro e os próximos serviços devem ser feitos baseados na atualização de como está a umidade nos arcos. A distância considerada é o número de períodos requeridos para atravessar uma aresta. Aplicações em irrigação podem ocorrer em minas ao céu aberto, florestas, jardins entre ruas ou calçadas (canteiros).

A frequência no PCARP-IC é dinâmica, ou seja, as frequências variam conforme o horizonte de tempo, tendo-se níveis de umidade para cada horário e aresta. Além disso, é estipulado um nível de umidade desejado e um nível máximo. A água evapora e o tempo de evaporação depende do horário do dia e movimento de veículos, caracterizando um serviço irregular. Uma penalização na função objetivo deve ocorrer caso o nível desejado de umidade do arco esteja abaixo do desejado, sendo que o custo de penalização deve dominar sobre o custo de roteamento. Também é necessário levar em consideração diversos cenários, pois os níveis de umidade podem ser diferentes em cada via sempre que um modelo é iniciado.

A primeira variante desse problema traz uma proposta contando com um depósito de água e um veículo que pode ser reabastecido a qualquer momento no depósito (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014b). A velocidade

é constante, independente se está servindo ou apenas atravessando o arco. Dois cenários são considerados, nos quais a quantidade de água despejada pode ser fixa ou variável e um modelo matemático é proposto para cada cenário. Além disso, é apresentada uma solução para trabalhar com tempos diferentes de atravessamento e serviço, porém não é utilizada. São criadas instâncias a partir das 23 de Golden, Dearmos e Baker (1983) e outras 10 chamadas *jpr* que tem estrutura de árvores, semelhantes às vias de minas ao céu aberto. A resolução é feita usando o CPLEX 12.4 avaliando-se até quantos períodos era possível limitar o horizonte de tempo. Em geral, instâncias de até 22 nós, entre 40-55 arestas e com um horizonte de tempo variando entre 20-30, eram resolvidas. Usando um critério de parada com um gap de 2% resultavam em uma redução de 2-5% do tempo computacional sem perda de otimalidade.

Mais de um carro é usado em Riquelme-Rodríguez, Gamache e Langevin (2014a). Outra modelagem matemática é apresentada junto com uma heurística baseada em vizinhança chamada *Adaptive Large Neighborhood Search* (ALNS). A ALNS é uma heurística de remoção e inserção que permite alcançar uma grande vizinhança. Para sua aplicação foi necessária a obtenção de uma solução inicial por meio de uma heurística simples, no caso usou-se primeiramente um algoritmo de clusterização e depois de roteirização. Essa solução inicial encontrada foi melhorada pelo ALNS usando operadores de destruição e reparação detalhados no trabalho. Foram criadas 22 novas instâncias, baseadas no trabalho de Golden e Wong (1981), pois uma requeria-se um horizonte de tempo muito grande, e outras 10, chamadas mines, baseadas em 5 minas ao céu aberto com 3 ou 5 veículos. Instâncias pequenas eram tratáveis com a formulação matemática, porém o foco foi na aplicação do ALNS programado em Python que acarretava em pequenas melhorias, entre 0% e 10% do valor das soluções iniciais que já eram boas.

A determinação da localização de depósitos que servem como pontos de retil para o PCARP-IC também é abordada neste trabalho, sendo nomeado de Problema de Localização em Roteamento em Arcos com Restrições de Estoque (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2016). Neste artigo, considera-se que os depósitos podem trabalhar com qualquer número de veículos e é dada importância a determinação da localização do depósito, pois como os veículos tem capacidade limitada, muitas vezes é necessário reabastecer os seus reservatórios de água. A velocidade servindo ou apenas atravessando é a considerada a mesma,

mesmo sabendo que a velocidade seja maior quando o veículo esteja apenas atravessando ou que ainda pode sofrer a interferência da presença de outros caminhões e da qualidade das vias. A função objetivo visa minimizar: o número de punições caso algum arco fique com o nível de umidade abaixo do desejado; os custos de transporte, sendo eles de serviço ou de atravessamento; o custo relativo a quantidade de caminhões usados; e o custo de estabelecer depósitos. O trabalho apresenta sua formulação matemática, dois algoritmos de localização e um algoritmo de alocação e roteirização de veículos, gerando soluções iniciais que são melhoradas de duas formas: um algoritmo que muda a localização inicial e o ALNS.

Pontos de refil também são abordados no trabalho de Huang e Lin (2014). É um trabalho chamado pelos autores de Problema de Roteamento Periódico em Arcos com Pontos de Refil. O problema é aplicado na irrigação de árvores na beira de estradas ou ruas, considerando que diferentes tipos de plantas requerem diferentes quantidades de água ao serem irrigadas e que uma quantidade em excesso ou muito pouco pode afetar a vida das árvores. Cada veículo pode retornar ao depósito ou parar em um posto para reabastecer seu tanque de água. É proposto um modelo matemático em que existe uma quantidade de carros pré-estabelecidas, a menor quantidade deles deve ser usada e os arcos com demandas devem ser atendidos quantas vezes forem necessárias. Com os pontos de refil sendo conhecidos, o problema é convertido em um Problema de Roteamento em Nós para ser resolvido com um algoritmo de colônia de formigas e de busca local. Dois conjuntos de instâncias são gerados a partir de Belenguer e Benavent (2003) e de Golden, Dearmon e Baker (1983) e um estudo de caso é feito para a cidade de Kaohsiung, em Taiwan, onde os algoritmos foram aplicados.

A mais recente abordagem trata de incerteza nos parâmetros de demanda, demandas estocásticas, formando o PCARP Robusto (TIRKOLAEI; MAHDAVI; MEHDI SEYYED ESFAHANI, 2018). Tem um enfoque no problema de coleta de lixo considerando múltiplas viagens e o tempo disponível dos trabalhadores para determinar o número de veículos. Visa minimizar a distância total percorrida e o custo de uso dos carros ao longo do horizonte de tempo em um grafo assimétrico. O problema é resolvido utilizando um algoritmo híbrido baseado em uma heurística construtiva e *Simulated Annealing*, algumas instâncias foram geradas aleatoriamente e também resolvidas pelo CPLEX. Uma das principais decisões do PCARP não é tomada nesse problema, todos os arcos requeridos devem ser

atendidos todos os dias, ou seja, não é necessário determinar quando atender cada demanda. Ele é chamado de periódico pelo fato de que a demanda flutua com uma certa incerteza ao longo do tempo, sendo assim as rotas são diferentes a cada dia.

Todos os trabalhos citam aplicações, porém alguns tem como foco principal o desenvolvimento de técnicas de resolução. Normalmente os trabalhos pioneiros têm geração de instâncias, modelagens matemáticas inéditas ou aplicação em dados reais. Outros trabalhos têm foco em abordagens apenas teóricas ou resolutivas, utilizando instâncias geradas pelos outros problemas, como os casos de Kansou e Yassine (2009), Mei *et al.* (2011), Zhang *et al.* (2016) e Chen e Hao (2018).

Uma proposta de PCARP Misto, utilizando um grafo misto, dois custos diferentes por *link* (com tratamento e sem) e arestas com vento (custos dependentes da direção) é feita no trabalho de Kansou e Yassine (2009), porém essas propostas não interferem nas estruturas propostas até em então de PCARP e não agregam tanto ao desenvolvimento do tema. A maior contribuição de Kansou e Yassine (2009) é a aplicação de um algoritmo de colônia de formigas combinado com uma heurística de inserção implementado em C e usando as instâncias *pgdb* com sucesso (13 dos 23 melhores resultados encontrados na literatura até então).

Algoritmos Meméticos são aplicados ao PCARP em Mei *et al.* (2011) e Zhang *et al.* (2016) que incluem boas soluções em suas populações iniciais vindas de outros algoritmos como *path scanning*. Claramente, ambas as pesquisas trabalham com o objetivo primário de reduzir o número de veículos, visto que isso afetará todas as designações de tarefas para rotas, além do próprio custo das rotas; sendo o objetivo secundário a minimização do custo total com viagens.

Em ambos os trabalhos foram alcançados resultados muito bons em instâncias consagradas da literatura. Até então, os algoritmos utilizados em trabalhos (CHU; LABADI; PRINS, 2006; LACOMME; PRINS; RAMDANE-CHÉRIF, 2005) eram baseados nas técnicas para o CARP, mais especificamente com algoritmos evolutivos com presença de operadores de *crossover*. Como operações de *crossover* geram filhos que herdam características dos pais, dificilmente o número de veículos utilizados é reduzido.

Mei *et al.* (2011) ressaltam que a representação das soluções é algo fundamental no desenvolvimento de uma heurística estocástica, como ocorre com os Algoritmos Meméticos. Neste sentido, apresentam uma representação

diferenciada das tradicionais, que eram baseadas em sequências de atividades. Essa nova representação mostra de maneira mais clara a rota que será feita, tornando o processo de busca local mais eficiente. A proposta é de um algoritmo que combina fusão de rotas com o Algoritmo Memético, *Memetic Algorithm With Route-Mergin* (MARM). A fusão é feita de forma que algumas rotas deixam de existir e a suas tarefas são redistribuídas dentro das demais rotas. Seu algoritmo é aplicado nas instâncias *pgdb* e *pval*, além de outras 10 instâncias que representam problemas reais, sendo elas compostas por duas estruturas de grafos com 5 capacidades diferentes obtidas de modificações de Brandão e Eglese (2008), chamadas de *pG*. Soluções melhores foram alcançadas para todas as 57 instâncias disponíveis na literatura quando avaliado o número de veículos utilizados nas frotas e em 56 delas os custos com deslocamento eram melhores, com tempos computacionais também em geral melhores.

Já em Zhang *et al.* (2016) é desenvolvido um Algoritmo Memético com decomposição de rotas para o ambiente do PCARP, *Memetic Algorithm with Route Decomposing* (MARD). Alguns autores são os mesmos do trabalho de Mei *et al.* (2011), de modo que as ideias são similares. O MARD melhora simultaneamente os objetivos de minimizar o número de veículos e os custos com rotas. Eles afirmam que “a ideia básica é selecionar uma rota do período que tiver o maior número de rotas e a decompor. Durante a decomposição, cada tarefa na rota selecionada é movida para uma posição em outra rota que acarrete no menor custo adicional” (ZHANG *et al.*, 2016, p. 4, tradução nossa). O algoritmo foi aplicado com sucesso nas instâncias *pgdb*, *pval* e *pG*, além de reimplementar alguns algoritmos para comparação dos resultados.

Os autores Chen e Hao (2018) também tratam o PCARP de forma similar a Mei *et al.* (2011) e Zhang *et al.* (2016). Dessa vez o algoritmo proposto é um híbrido de busca local que opera em duas fases, sendo que a primeira fase visa reduzir o número de veículos enquanto a segunda fase usa o resultado da primeira para como limitador para o problema como um todo. Até a data da submissão o artigo de Zhang *et al.* (2016) ainda não estava disponível e os resultados foram comparados com os de Mei *et al.* (2011). As duas fases integram operadores de busca e heurísticas trazendo bons resultados para 63 instâncias compostas por *pgdb*, *pval* e *pG*, sendo que em 44 foram conhecidos novos melhores resultados.

2.2.2 Proposta taxonômica para o PCARP

Uma proposta taxonômica permite uma descrição sistematizada de algum problema e, até o momento, nenhuma proposta desse nível foi feita para o PCARP, apenas uma simples classificação foi apresentada em Lacomme, Prins e Randane-Chérif (2005). Essa classificação já havia sido abordada previamente em conferências (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, 2002c) e foi incrementada posteriormente com a classificação de serviços irregulares e regulares (MONROY; AMAYA; LANGEVIN, 2013). A classificação taxonômica dos trabalhos é algo recorrente na abordagem de outros temas de Pesquisa Operacional, inclusive quando se trata de roteamento de veículos, como no caso do *Inventory Routing Problem* (IRP) (ANDERSSON *et al.*, 2010; GUIMARÃES, 2015), do *Location Routing Problem* (LRP) (LOPES, RUI BORGES; FERREIRA; SOUSA, 2013) e do Problema de Roteamento de Veículos (EKSIOGLU; VURAL; REISMAN, 2009). A classificação taxonômica pode levantar vários aspectos dos problemas estudados. Dois níveis são usados em Lopes, Ferreira e Sousa (2013), levando em consideração aspectos físicos do problema e, depois, a abordagem de resolução adotada.

A classificação taxonômica do PCARP aqui apresentada irá abordar os seguintes aspectos dos trabalhos acadêmicos: período, horizonte de tempo, natureza do serviço, roteirização, dimensão da frota, frequências, demanda, rigor da demanda, natureza do grafo, foco e método de resolução. Maiores detalhes das aplicações e abordagens de resolução são apresentados no decorrer deste trabalho, bem como a classificação dos trabalhos segundo a taxonomia proposta.

Para o estudo em questão, fez-se uma busca nos portais de pesquisa/bases de dados Web-of-science e Scopus, com as palavras chave: “*Periodic Capacitated Arc Routing Problem*” e “*Periodic Arc Routing*”. Após um processo de filtragem dos artigos, que consistiu em verificar se o assunto abordado se relaciona com a pesquisa aqui proposta, foram encontrados 13 artigos em cada uma das plataformas. Quando mesclados os dois resultados obteve-se o resultado de 14 artigos.

A partir desses 14 artigos, verificaram-se as referências bibliográficas que tratam do assunto. Foram encontrados 5 trabalhos que não estavam na base de dados. Verificou-se, também, que existem outros artigos publicados em anais de congressos/simpósios que não estão acessíveis e, por este motivo, não compõe

parte da lista de análise. Além desses trabalhos, consideraram-se os 2 artigos publicados em revistas pelo próprio autor deste. O QUADRO 2.1 resume a lista de trabalhos que servem como base para classificação taxonômica dos problemas.

QUADRO 2.1 - TRABALHOS ACADÊMICOS ACERCA DO PCARP.

Trabalhos listados da base de dados <i>web-of-science</i> e <i>scopus</i>	(CHEN; HAO, 2018); (TIRKOLAEI; MAHDAVI; MEHDI SEYYED ESFAHANI, 2018); (ZHANG <i>et al.</i> , 2016); (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2016); (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014b); (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a); (HUANG; LIN, 2014); (MONROY; AMAYA; LANGEVIN, 2013); (MEI <i>et al.</i> , 2011); (CHU; LABADI; PRINS, 2006); (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005); (CHU; LABADI; PRINS, 2005); (GHIANI <i>et al.</i> , 2005); (CHU; LABADI; PRINS, 2004c)
Artigos citados nas referências	(LACOMME; PRINS; RAMDANE-CHÉRIF, 2002c); (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a); (CHU; LABADI; PRINS, 2003); (MARZOLF; TRÉPANIER; LANGEVIN, 2006); (KANSOU; YASSINE, 2009);
Artigos do autor	(BATISTA; SCARPIN, 2014); (BATISTA; SCARPIN, 2015);

FONTE: O Autor (2018).

2.2.2.1 Período

O período foi a primeira característica usada para classificar o PCARP, dividindo-se o problema em duas classes, A e B (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, 2002c, 2005). Porém, neste trabalho será atribuído um nome representativo à classificação dos períodos e as classes A e B serão desconstruídas por incompatibilidade com outros trabalhos.

A Classe A abrange problemas nos quais a demanda de um arco não resulta de uma acumulação de produção diária. O período em que o serviço é executado não depende da demanda e dos custos, isto é, “a demanda de cada arco (e seu custo de serviço) serão os mesmos para qualquer dia de serviço” (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, 2002c, 2005, tradução nossa). A partir dessa definição da Classe A, aproveitar-se-á apenas a característica de que o período de

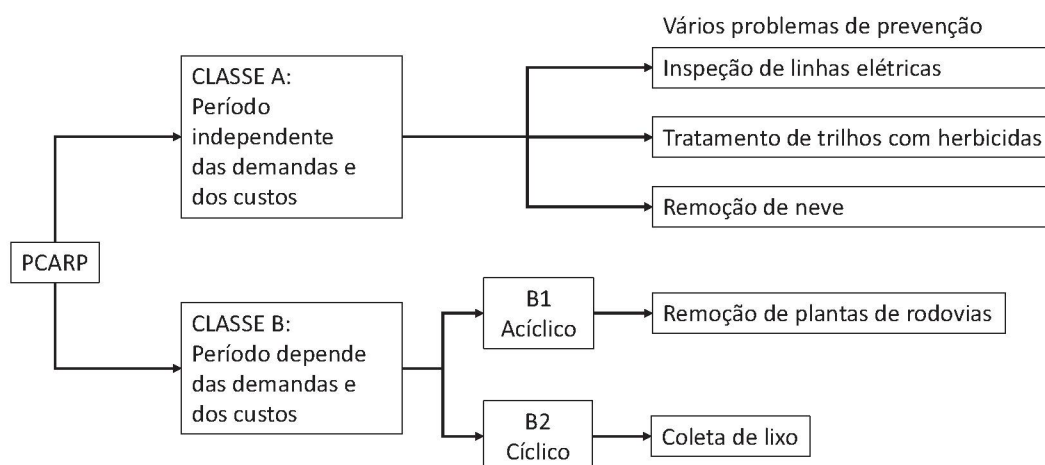
atendimento é independente das demandas e dos custos. O fato da demanda e custo serem os mesmos para qualquer dia de serviço será explorado como característica do serviço, que foi proposta em Monroy, Amaya e Langevin (2013). Exemplos de períodos independentes estão ligados a problemas preventivos como tratamento de trilhos com herbicidas, inspeção de linhas de energia elétrica e problemas de monitoramento. Uma frequência é o suficiente para determinar quando esses serviços devem ocorrer.

A independência está relacionada com o modo como o problema está sendo abordado dentro dos trabalhos acadêmicos. A independência muitas vezes é uma simplificação para os problemas em que o período de atendimento depende de uma acumulação de fatores difíceis de serem mensurados. Por exemplo, no problema de inspeção em trilhos de trem, cada arco tem sua periodicidade, porém essa periodicidade é resultado da quantidade de trens que passa em cada trecho, determinando um maior ou menor gasto do trilho, interferindo nos períodos de atendimento. Outro exemplo pode ser relacionado a inspeção de linhas elétricas, se algumas linhas estão mais expostas a piores condições climáticas, elas provavelmente deverão ser analisadas com maior frequência.

A Classe B engloba problemas em que o período de atendimento do arco é influenciado por uma produção diária que ocorre continuamente, ou ainda, o período depende da demanda e dos custos. Por exemplo, a demanda da coleta de lixo é cada vez maior sempre que o lixo não é coletado, pois a produção de lixo é contínua. A Classe B citada é subdividida em problemas cíclicos e acíclicos. Apesar dessa subdivisão estar associada apenas a essa classe de problemas, utilizar-se-á essa subclassificação para classificar os problemas gerais de PCARP, sendo detalhada na próxima seção, sobre horizonte de tempo. A FIGURA 2.1 apresenta a classificação proposta em Lacomme, Prins e Ramdane-Chérif (2002c, a, 2005).

O período pode ser classificado como dependente ou independente. Independente para os problemas que já tem frequências pré-estabelecidas e nos quais as demandas não sofrem um acúmulo, ou seja, todos os fatores que influenciam no atendimento de um arco estão implícitos. Já os dependentes são usados quando a demanda sofre um acúmulo devido ao decorrer do tempo e isso interfere diretamente na resolução do problema; por exemplo, um caminhão de lixo receberá uma carga maior quanto maior for o espaço entre coletas, influenciando na utilização de sua capacidade.

FIGURA 2.1 - UMA SIMPLES CLASSIFICAÇÃO DO PCARP



FONTE: Adaptada de Lacomme, Prins e Ramdane-Chérif (2005).

2.2.2.2 Horizonte de tempo

Toda resolução de problema relacionado com o PCARP possui um início e um fim bem definidos. Porém, o problema pode ser caracterizado como cíclico ou acíclico, caso da Classe B, FIGURA 2.1, que aqui será estendida para qualquer tipo de PCARP. A extensão é feita porque podem existir problemas em que o período independente pode ser classificado das duas formas. O Problema de Roteirização de Estoques (PRE) classifica o horizonte de tempo como sendo infinito ou finito (ANDERSSON *et al.*, 2010; GUIMARÃES, 2015). A proposta do horizonte de tempo é definida da seguinte forma:

“A classificação quanto ao horizonte de tempo envolve a possibilidade de uma quantidade finita ou infinita de períodos. Um horizonte finito implica na resolução do PRE em um espaço limitado de tempo, como um ano, um mês ou uma semana, onde a fração temporal pode ser mensal, semanal ou diária. Como exemplo, tem-se 30 períodos em um horizonte mensal finito, para o caso de se planejar entregas diárias em um dado mês. Já o horizonte temporal infinito está associado a planejamento com padrões de entrega que se repetem, mas não estão necessariamente limitados por um período mais extenso. Tais padrões constituem o chamado horizonte rolante que pode ser indefinidamente deslocado. Um exemplo para esta situação seria programar entregas semanais periódicas padronizadas (mesmos clientes, mesmas quantidades e mesmas rotas) ao longo de meses ou anos.” (GUIMARÃES, 2015, p. 35)

A classificação de problemas cíclicos ou acíclicos, está relacionada com o horizonte de tempo e necessita ser bem definida, algumas das definições já propostas chegam a se contradizer. Por exemplo, em um problema acíclico “o

horizonte de tempo é não cíclico e a demanda $r(u)$ de uma tarefa u em um dia p é a produção acumulada desde o dia de serviço anterior” (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, p. 3, tradução nossa). Entretanto, o problema da coleta de lixo se encaixaria nessa definição e é classificado como cíclico, logo depois justificado pelo fato de que as produções de lixo se repetem semanalmente. Em Lacomme, Prins e Ramdane-Chérif (2005, p. 540) a definição sofre uma alteração para: “o horizonte é linear: seu início e fim é bem identificado”, porém tanto os problemas cíclicos quanto acíclicos quando modelados têm um início e fim bem definidos, não sendo uma característica capaz de defini-los.

Desta forma, a classificação como acíclica será, neste trabalho, relacionada com horizonte finito. O início e fim são bem definidos e as decisões de roteamento tomadas servem apenas para o período estipulado.

Sendo H o horizonte de tempo, a classificação de problemas da Classe B2, cíclicos, da FIGURA 2.1, é dada por: “ H se repete, mas os dias de serviço selecionados devem ser idênticos em todas as repetições” (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005, p. 540). Os problemas cíclicos estarão relacionados com um horizonte de tempo infinito. A definição de problemas cíclicos é baseada no fato de que o horizonte de tempo se repete indefinidamente, de modo que as decisões de roteamento devem ser idênticas para cada horizonte de tempo. Tem-se então um horizonte de tempo rolante que pode ser deslocado quantas vezes forem necessárias.

A classificação de cíclica ou acíclica foi dada logo no início dos estudos sobre o PCARP (LACOMME; PRINS; RAMDANE-CHÉRIF, 2002a, 2002c, 2005), porém poucos trabalhos classificam essa característica, ainda que citem esses trabalhos. A partir daqui todo trabalho em que é possível replicar o horizonte de tempo será classificado como cíclico. Por exemplo, no caso de monitoramento de rodovias (MONROY; AMAYA; LANGEVIN, 2013) os planejamentos feitos são em geral para duas semanas e o trabalho não impõe replicações futuras. Seu planejamento poderia se repetir muitas vezes, mesmo sendo imaginável que existam temporadas de maior movimento nas rodovias, quando o planejamento deveria mudar. Caso esse fosse o cenário, uma solução poderia ser a utilização de um horizonte de planejamento que contemple todas as temporadas e que se repetiria indefinidamente. Existem outros trabalhos em que a mesma solução não é válida

para períodos futuros, como em Riquelme-Rodríguez, Gamache e Langevin (2014a, 2016).

2.2.2.3 Serviços

Todo PCARP consiste em atendimentos a arcos, ou melhor, serviços ou tarefas que devem ser executados durante uma travessia. O trabalho de Monroy, Amaya e Langevin (2013) se apropria da classificação dos períodos independentes e dependentes proposta em Lacomme, Prins, Ramdane-Chérif (2002c, a, 2005), porém abandona a subclassificação de problemas acíclicos e cíclicos e define conceitos de serviços regulares e irregulares para as classes A e B da FIGURA 2.1. As características propostas serão aqui também usadas para classificar o PCARP.

Para essa classificação é importante a definição de subperíodos, lembrando que um horizonte de tempo é formado por períodos e os subconjuntos compostos por períodos são chamados de subperíodos. Por exemplo, um horizonte de tempo de uma semana pode ser subdivido em dois subperíodos: dias úteis e final de semana. Ou ainda, uma semana pode ser dividida em sete subperíodos onde cada um representa um dia da semana.

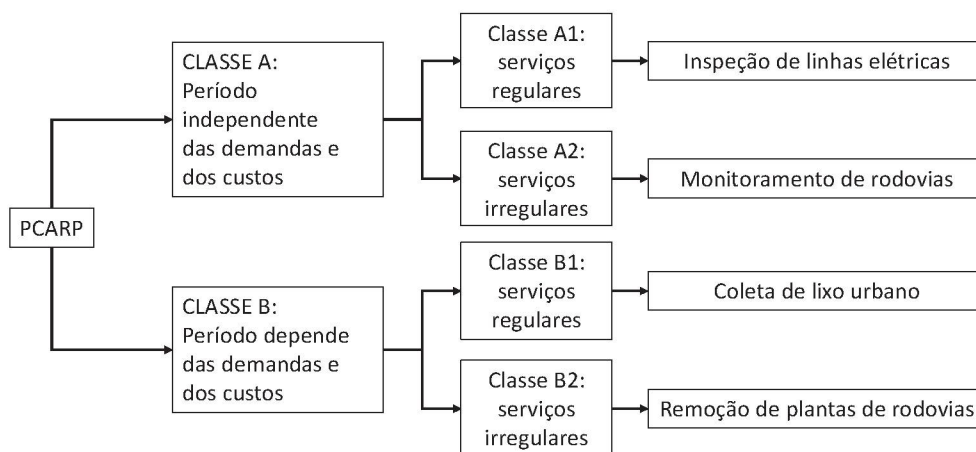
Os serviços regulares estão associados a uma única frequência que engloba todo o horizonte de tempo, ou ainda, uma mesma frequência para cada subperíodo desde que todos os subperíodos tenham o mesmo tamanho. É também o caso de quando se tem as periodicidades bem definidas que nunca se alteram. Por exemplo, a coleta de lixo deve ser feita duas vezes na semana, ou um arco deve ser atendido a cada 3 dias. Os serviços irregulares ocorrem quando existe diferentes frequências para cada subperíodo, ou ainda, frequências iguais para subperíodos de tamanhos diferentes. Por exemplo, se considerado um horizonte de tempo de uma semana, sendo ela dividida em fim de semana e dias úteis, existe uma frequência distinta para cada uma dessas subdivisões.

A subclassificação feita em Monroy, Amaya e Langevin (2013) pode ser encontrada na FIGURA 2.2 e seu detalhamento está indicado logo a seguir. A subclassificação dentro da classificação dos períodos não é necessária, pois ela se repete para cada tipo de período, logo, ela deve ser tratada como uma classificação do PCARP. Neste sentido, observa-se que:

“Subclasses A1 e B1 consistem de problemas nos quais o planejamento deve ser executado ao longo de um horizonte de tempo sem subdivisões. A determinação de uma frequência única para cada tarefa sobre todo o horizonte de tempo deveria ser suficiente. Por exemplo, coleta de lixo normalmente é planejada para uma ou duas semanas e a produção de lixo de cada rua é traduzida em uma frequência de serviço sobre o horizonte de tempo (por exemplo, duas vezes na semana).”

Em contrapartida, em operações de monitoramento, poderia ser importante levar em conta diferentes frequências para algumas tarefas sobre o horizonte de tempo. Ruas principais podem requerer mais visitas durante os finais de semana por causa de um tráfego maior e menos durante os dias úteis. Nesse caso, a subdivisão de todo o horizonte de tempo é considerada para incluir essa característica. Subclasses A2 e B2 são ligadas ao horizonte de tempo envolvendo subperíodos. A frequência de serviços acerca das tarefas varia dependendo desses subperíodos.” (MONROY; AMAYA; LANGEVIN, 2013, p. 692, tradução nossa)

FIGURA 2.2 - CLASSIFICAÇÃO DE PROBLEMAS DE ROTEAMENTO EM ARCOS PERIÓDICOS



FONTE: Adaptada de Monroy, Amaya e Langevin (2013).

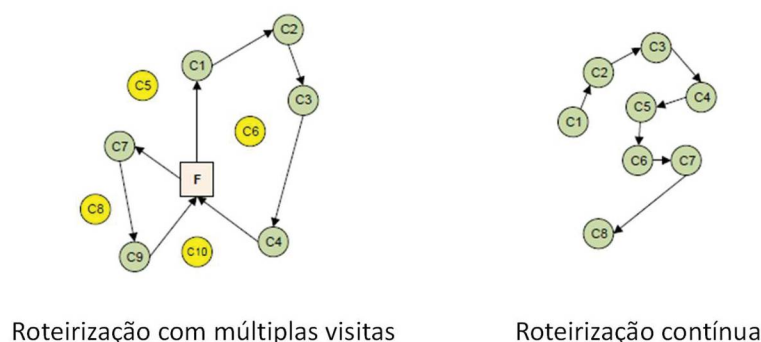
Diante disso, os trabalhos levantados serão classificados como contendo serviços regulares ou irregulares.

2.2.2.4 Roteirização

O Problema de Roteirização de Estoques (PRE) é um dos problemas que categoriza os trabalhos acadêmicos segundo a sua roteirização (ANDERSSON *et al.*, 2010; GUIMARÃES, 2015). No PRE a estrutura logística do problema diz como será o roteamento. Basicamente, ele pode ser abordado com um Problema de Roteamento de Veículos (PRV) ou como um Problema de Coleta e Entrega (Pick-up and Delivery Problem - PDP). A abordagem clássica dos problemas PRV utiliza-se do depósito como lugar de início e fim de rotas e como armazéns centrais desde

onde os produtos são distribuídos. Em contrapartida, existe o PDP, no qual não existe um depósito central e as rotas são feitas a partir de clientes para clientes – problema comum em casos marítimos – (ANDERSSON *et al.*, 2010). O roteamento é classificado de três formas no PRE: direta, múltiplas visitas e contínuo.

FIGURA 2.3 - TIPOS DE ROTEIRIZAÇÃO USADOS NO PCARP



FONTE: Adaptada de Guimarães (2015).

O Roteamento Direto não ocorre no PCARP, pois é o caso em que um fornecedor entrega seus produtos diretamente para um cliente e volta para seu ponto de partida. O Roteamento com múltiplas visitas é o roteamento mais comum abordado no PCARP, um veículo visita mais de um cliente por viagem começando e terminando a rota em um depósito. Já o PDP é caracterizado por sua rota ser contínua, sem início e sem fim, sem a obrigatoriedade de voltar a um depósito.

A roteirização do PCARP será subdividida em múltiplas visitas ou contínua. A FIGURA 2.3 adaptada de Guimarães (2015) ilustra como funcionam os dois tipos de roteirização para o PCARP: círculos representam clientes e quadrados o depósito desde onde saem os fornecedores, clientes em amarelo não são servidos pelas rotas. O PCARP contínuo é abordado apenas em Batista (2014) e Batista e Scarpin (2014, 2015), porém não havia sido classificado dessa forma até então. Todos os demais trabalhos são classificados como múltiplas visitas.

2.2.2.5 Dimensão da frota

A dimensão da frota visa classificar o PCARP em virtude da quantidade de veículos disponíveis para a resolução do problema, podendo ser de três maneiras diferentes: um veículo, múltiplos veículos ou irrestrito. Um veículo se refere à frota composta por apenas um veículo que deve atender todos os arcos requeridos. Em

múltiplos veículos, frotas compostas por mais de um veículo devem atender as demandas da melhor maneira possível, sendo eles usados ou não. Irrestrito é caracterizado pelo problema em que o tamanho da frota é desconhecido a priori e deseja-se minimizá-lo, ou seja, descobrir o número mínimo de veículos para resolver o problema em questão.

2.2.2.6 Frequências

A frequência é uma característica marcante em todos os trabalhos, pois sua definição é determinante da forma como o problema será abordado, as vezes ela torna a resolução do problema mais simples, porém limita o espaço solução. A resolução do PCARP básico parte do pressuposto de que as decisões estratégicas já foram tomadas e a forma com que as frequências são traduzidas para os modelos podem afetar as decisões táticas e operacionais (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005). A demanda influencia completamente a frequência e acaba, muitas vezes, dependendo do período ou da data do atendimento anterior. Classificou-se quatro formas distintas de frequência já abordadas nos trabalhos: pura, combinação de dias, periodicidade e dinâmica.

A frequência ou frequência de serviço é a quantidade de vezes que uma tarefa deve ser executada ao longo do horizonte de tempo. A frequência pura segue essa definição de frequência, lembrando que também pode ser atribuída a um subperíodo. Ou seja, ela é um número associado a um intervalo de tempo que pode ser menor ou igual ao horizonte de tempo. Em Monroy, Amaya e Langevin (2013), as demandas são associadas aos subperíodos criados na forma de frequência, por exemplo, um atendimento durante os dias úteis e dois no final de semana. Nesse caso específico os atendimentos devem ser maximizados, ou seja, ocorrerem em quantidades maiores ou iguais que as frequências. A frequência total de cada arco pode ser obtida mediante a soma de todas as frequências associadas aos subperíodos do respectivo arco.

Alguns dos trabalhos utilizam restrições referentes ao espaço de tempo entre serviços que podem ser basicamente duas: combinação de dias e periodicidade. A forma da combinação de dias é a mais utilizada quando trata-se do PCARP, isso pelo fato de existirem duas grandes vantagens: um espaço de tempo entre atendimentos é implicitamente satisfeito e os cálculos de demanda podem ser

antecipados (CHU; LABADI; PRINS, 2005; LACOMME; PRINS; RAMDANE-CHÉRIF, 2002c, a).

O funcionamento da combinação de dias consiste em selecionar, para cada arco requerido (tarefa), uma combinação de dias de atendimento de forma que as demandas sejam atendidas. Cada tarefa (u) é associada a uma lista $comb(u)$ que contém combinações de dias em que seria possível fazer os atendimentos. Por exemplo, dado um horizonte de tempo de uma semana (7 dias) para a coleta de lixo, uma combinação de atendimentos possível para uma determinada rua, tarefa (u), poderia ser de coletas nas segundas, quartas e sextas ou terças, quintas e sábado, isso geraria 2 itens para a lista $comb(u)$. O problema então consiste na designação de uma combinação de dias para cada tarefa. Logo após resolve-se o roteamento para cada dia onde as tarefas previstas (selecionadas por meio da designação) são atendidas.

As restrições que usam intervalos de tempo para cada tarefa estão relacionadas à periodicidade, o período previsto entre dois acontecimentos sucessivos. A frequência tratada como intervalos de tempo ou periodicidade consiste na definição de um mínimo espaço de tempo entre a execução de cada tarefa e/ou um máximo espaço de tempo. A utilização da periodicidade forma um problema mais complexo de ser resolvido do que quando se tem a combinação de dias. Isso acontece devido ao fato de que na combinação de dias o espaço solução dos problemas acaba sendo limitado porque os dias são previamente selecionados.

Por fim, a mais recente abordagem da frequência na literatura é uma abordagem dinâmica. A frequência dinâmica não pré-define espaços entre serviços, nem uma quantidade de vezes necessária para atender cada arco. As demandas acabam de uma forma ou de outra dependendo sempre do período de atendimento: quanto mais tempo um arco demora para ser atendido maior é a sua demanda. A frequência tratada de forma dinâmica é estabelecida como um resultado do problema resolvido, ou seja, a solução do próprio modelo indicará quantas vezes um arco deve ser atendido. A frequência em sua definição não é usada, mas obtida através das necessidades, ou melhor, as frequências variam conforme o tamanho do horizonte de tempo e os critérios que definem a demanda. Essa abordagem é usada em Riquelme-Rodríguez, Gamache e Langevin (2014a, 2014b, 2016).

2.2.2.7 Demanda

A demanda é uma das características mais influentes do PCARP, “na prática, custos de serviços são derivados das demandas ... a demanda (e consequentemente, o custo de serviço) frequentemente dependem do período” (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005, p. 539, tradução nossa). Em outras classes de problemas de roteamento, assim como aqui, a demanda é classificada como: estocástica ou determinística. Na estocástica a demanda é incerta e se altera conforme uma probabilidade, na prática quase todos os problemas possuem essa característica, porém não são tratados como tais.

Recentemente foi publicado o primeiro trabalho tratando de incertezas na demanda, ou seja, demandas estocásticas para o PCARP recebendo o nome de PCARP Robusto (TIRKOLAEI; MAHDAVI; MEHDI SEYYED ESFAHANI, 2018). Todos os demais trabalhos publicados sobre PCARP possuem a demanda abordada de forma determinística, porque por mais que ela se altere ao longo do horizonte de tempo, sempre pode ser calculada e conhecida a priori. Relembrando que a demanda interfere completamente nos períodos e na maneira com que as frequências são estabelecidas.

2.2.2.8 Rigor da demanda

Os problemas podem tentar minimizar a quantidade de carros ou ter a quantidade de carros pré-estabelecida, porém sempre tentam ocupar as suas capacidades da melhor forma. Quando um carro está com sua capacidade em 100% em um dado período significa que ele não pode mais fazer atendimentos nesse período determinado, logo seria necessária uma frota maior. Porém, existem casos em que o atraso do atendimento de uma tarefa poderia ser benéfico ao aproveitar melhor a quantidade de carros utilizada, postergando um atendimento.

Algumas abordagens permitem que um serviço seja atrasado ao custo de uma penalidade na função objetivo, para esses estabelece-se o rigor da demanda como flexível. Em outras abordagens não é permitido que ocorra atraso algum, para esses diz-se que o rigor da demanda é rígido, isso pode ocorrer até mesmo implicitamente quando a frequência é estabelecida como uma combinação de dias.

2.2.2.9 Grafo

Outro fator que interfere na criação de um modelo que representa a realidade é o grafo. Nesta proposta taxonômica, eles podem ser direcionados ou não direcionados, também chamados de orientados ou não orientados, respectivamente. Alguns autores utilizam uma terminologia específica para arcos e arestas (CORMEN *et al.*, 2002), considerando que arcos são ligações que tem sentido e arestas não. Entretanto, no âmbito desta pesquisa não cabem estas diferenciações entre arco e aresta, uma vez que esta diferença não está clara na maioria dos trabalhos localizados.

Um grafo não direcionado pode modelar apenas vias de mão dupla, de modo que toda tarefa pode ser feita em qualquer direção, inclusive se for necessário atendimentos dos dois lados da via (coleta bilateral) (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005). Em contrapartida, muitos problemas precisam ser modelados com grafos direcionados como no caso da coleta de lixo em vias de mão única.

A classificação de grafos poderia ser estendida para outros tópicos, como os multigrafos, nos quais podem existir arestas paralelas que retratariam melhor alguns problemas. Isso não será feito, pois a modelagem final sempre acaba recaindo na discussão de ser direcionado ou não. Em geral, quando é trabalhado com problemas não direcionados, cada aresta é convertida em dois arcos com direções opostas. Muitos dos trabalhos ressaltam que podem ser aplicados para qualquer tipo de grafo, porém usam grafos não direcionados em seus testes. Há também um trabalho que faz referência aos grafos mistos (KANSOU; YASSINE, 2009), combinando direcionados e não, que para fins de classificação nesta proposta taxonômica será considerado como direcionado.

2.2.2.10 Foco

Os trabalhos podem ter basicamente dois focos: aplicável ou teórico. Em geral todos os trabalhos citam aplicações, pois elas atribuem sentido ao desenvolvimento das pesquisas e exemplificam bem os propósitos de modelos matemático, suas restrições e variáveis. O critério para classificação se sustenta no fato de que em algumas pesquisas, o foco principal é dado a alguma aplicação, a

qual é bem descrita e acaba gerando novos modelos e novas formas de resolução. Estes trabalhos serão classificados como foco aplicável. Em contrapartida, mas não menos importante, os trabalhos classificados como teóricos, acabam tendo foco puramente teórico, baseados em fundamentações e provas matemáticas, técnicas de resolução e comparações de modelos já apresentados.

2.2.2.11 Abordagem de resolução

Todo trabalho visa alcançar resultados que sejam representativos para suas aplicações ou modelos, no entanto a maneira de alcançar esses resultados diferencia um trabalho do outro. Três formas de abordagem de resolução serão usadas para classificar os trabalhos: exata, heurística e ambas.

Os trabalhos que serão classificados com abordagem exata são os que basicamente respeitam completamente às formulações matemáticas e são resolvidos por meio de algum método exato ou algoritmo iterativo que faz a busca da solução ótima, podendo alcançá-la ou não. No caso de não encontrar a solução ótima, é apresentado o resultado obtido após um limite de tempo, essa solução é chamada de incumbente sendo a melhor encontrada até aquele momento. Além disso, outros trabalhos classificados como exatos são os que visam fazer provas matemáticas para definir, por exemplo, o limite inferior teórico da solução, baseado em conceitos de dualidade da programação linear de um problema.

Outros trabalhos chegam a desenvolver modelos, ou parte deles, sem tentar resolvê-los de forma exata, abordando-os exclusivamente de forma heurística, sendo desta forma classificados. A abordagem heurística conta com algum algoritmo que busca soluções boas, porém não garantem otimalidade (PEARL, 1984). Problemas que são subdivididos e resolvidos de maneira exata em cada uma de suas subdivisões são classificados como heurísticos, pois, não se encontra, necessariamente, a melhor solução para o problema com a soma de resultados ótimos das subdivisões.

A classificação de “ambas” ocorre quando em um mesmo trabalho é proposto a formulação matemática, resolvida por algum algoritmo exato, e também por métodos heurísticos. Muitas vezes apenas instâncias muito pequenas podem ser resolvidas de forma exata e, nestes casos, as heurísticas são propostas para problemas maiores, estes trabalhos também são classificados como “ambas”.

2.2.3 Resumo da Proposta Taxonômica

QUADRO 2.2 - RESUMO DA PROPOSTA TAXONÔMICA

Característica	Possibilidades
Período	Dependente (D); Independente (I).
Horizonte de tempo	Cíclico (C); Acíclico (A).
Serviço	Regular (R); Irregular (I).
Roteirização	Múltipla (M); Contínua (C).
Dimensão da frota	Um veículo (U); Múltiplos veículos (M); Irrestrito (I).
Frequências	Simples (S); Intervalo de tempo (I); Combinação de dias (C); Dinâmica (D).
Rigor da demanda	Flexível (F); Rígida (R).
Grafo	Direcionado (D); Não-direcionado (ND).
Foco	Teórico (T); Aplicado (A).
Abordagem de resolução	Exato (E); Heurístico (H); Ambos (A).

FONTE: O Autor (2018).

O QUADRO 2.2 apresenta um resumo de todas as possibilidades geradas na classificação taxonômica. O período sendo dependente ou independente; o horizonte de tempo cíclico ou acíclico; o serviço regular ou irregular; a roteirização múltipla ou contínua; a dimensão da frota como de um veículo, múltiplos ou irrestrita; o rigor da demanda como flexível ou rígida; grafo direcionado ou não; foco teórico ou aplicado; e a abordagem de resolução exata, heurística ou ambas.

2.2.4 Classificação Taxonômica dos Trabalhos

Essa seção classifica todos os trabalhos listados no QUADRO 2.1, bem como traz análises estatísticas das publicações em revistas.

Todos os trabalhos usados na criação taxonômica proposta serão classificados e estão listados no QUADRO 2.3. Lembrando que as classificações seguem as seguintes características: Período - Dependente (D) ou Independente (I); Horizonte de tempo- Cíclico (C) ou Acíclico (A); Serviço - Regular (R) ou Irregular (I); Roteirização - Múltipla (M) ou Contínua (C); Dimensão da frota - Um veículo (U), Múltiplos veículos (M) ou Irrestrito (I); Frequências - Simples (S), Intervalo de tempo (I), Combinação de dias (C) ou Dinâmica (D); Demanda - Determinística (D) ou Estocástica (E); Rigor da demanda - Flexível (F) ou Rígida (R); Grafo - Direcionado

(D); Não-direcionado (ND); Foco - Teórico (T) ou Aplicado (A); Abordagem de resolução - Exato (E); Heurístico (H); Ambos (A).

A classificação permite avaliar a diversidade de problemas abordados acerca do PCARP. Pode-se observar que existe uma riqueza grande de detalhes em torno do problema que geralmente acarretam em uma difícil resolução. Alguns padrões podem ser observados: a maior parte dos trabalhos tem demandas dependentes, trabalham com um horizonte de tempo cíclico, serviços regulares, roteirização múltipla, frota múltipla, frequência utilizando combinação de dias, com rigor de demanda rígido e grafos não direcionados.

QUADRO 2.3 - CLASSIFICAÇÃO TAXONÔMICA DOS TRABALHOS

Trabalho	Período	Horizonte de tempo	Serviços	Roteirização	Dimensão da frota	Frequências	Demanda	Rigor da demanda	Grafo	Foco	Abordagem da resolução
BATISTA, Este trabalho, 2018	I	C	R	C	M	I	D	F	ND	T	A
CHEN; HAO, 2018	D	C	R	M	I	C	D	R	ND	T	H
TRKOLAEI, MAHDAVI, MEHDISEYYED ESFAHANI, 2018	D	C	R	M	I	S	E	R	ND	A	A
ZHANG <i>et al.</i> , 2016	D	C	R	M	I	C	D	R	ND	T	H
R.-RODRÍGUEZ; GAMACHE; LANGEVIN, 2016	D	A	I	M	M	D	D	F	ND	A	H
BATISTA; SCARPIN, 2015	I	C	R	C	M	I	D	F	ND	A	E
R.-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014b	D	A	I	M	U	D	D	F	ND	A	A
R.-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a	D	A	I	M	M	D	D	F	ND	A	E
HUANG; LIN, 2014	I	C	R	M	M	S	D	R	ND	A	H
BATISTA; SCARPIN, 2014	I	C	R	C	M	I	D	F	ND	A	E
MONROY; AMAYA; LANGEVIN, 2013	I	C	I	M	M	S	D	R	D	A	A
MEI <i>et al.</i> , 2011	D	C	R	M	I	C	D	R	ND	T	H
KANSOU; YASSINE, 2009	D	C	R	M	M	C	D	R	D	T	H
MARZOLF; TRÉPANIER; LANGEVIN, 2006	I	A	I	M	U	S	D	R	D	A	H
CHU; LABADI; PRINS, 2006	D	C	R	M	M	C	D	R	ND	A	H
LACOMME; PRINS; RAMDANE-CHÉRIF, 2005	D	C	R	M	I	C	D	R	ND	A	H
GHIANI <i>et al.</i> , 2005	I	C	R	M	U	C	D	R	ND	T	H
CHU; LABADI; PRINS, 2005	D	C	R	M	M	C	D	R	ND	A	A
CHU; LABADI; PRINS, 2004	D	C	R	M	M	I	D	R	ND	A	A
CHU; LABADI; PRINS, 2003	D	C	R	M	M	C	D	R	ND	T	E
LACOMME; PRINS; RAMDANE-CHÉRIF, 2002b	D	C	R	M	I	C	D	R	ND	A	H
LACOMME; PRINS; RAMDANE-CHÉRIF, 2002c	D	C	R	M	I	C	D	R	ND	A	H

FONTE: O Autor (2018).

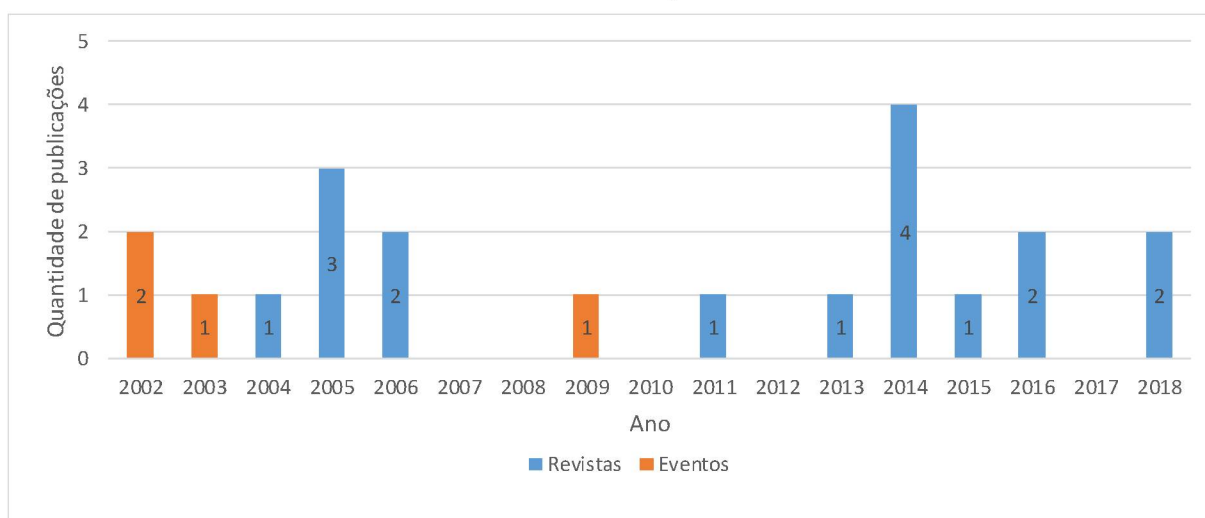
2.2.5 Bibliometria

Essa seção visa mostrar de forma simples a evolução da publicação dos trabalhos ao longo dos anos, ou seja, quais os autores que mais publicaram, as revistas que mais têm publicações e as origens segundo países dos trabalhos. Para algumas análises serão considerados apenas os artigos publicados em periódicos e para outras serão também considerados os trabalhos publicados em eventos que estão acessíveis online.

Os artigos que serão aqui abordados são os mesmos do QUADRO 2.1, lembrando que eles foram buscados em dois portais de pesquisa, levantados dentro das referências e adicionados dois artigos publicados pelo autor dessa tese.

O GRÁFICO 2.1 mostra a quantidade de artigos que foram publicados em cada ano, sendo eles de revistas ou em eventos. O ano que mais ocorreram publicações foi em 2014, o que demonstra que o assunto de fato é recente e tem sido abordado recorrentemente ao longo dos últimos anos. Se analisados os conteúdos dos artigos já citados, como exposto acima, pode-se verificar que o assunto tem se consolidado cada vez mais, com novas variantes e abordagens de resolução promissoras utilizando heurísticas.

GRÁFICO 2.1 - PUBLICAÇÕES POR ANO



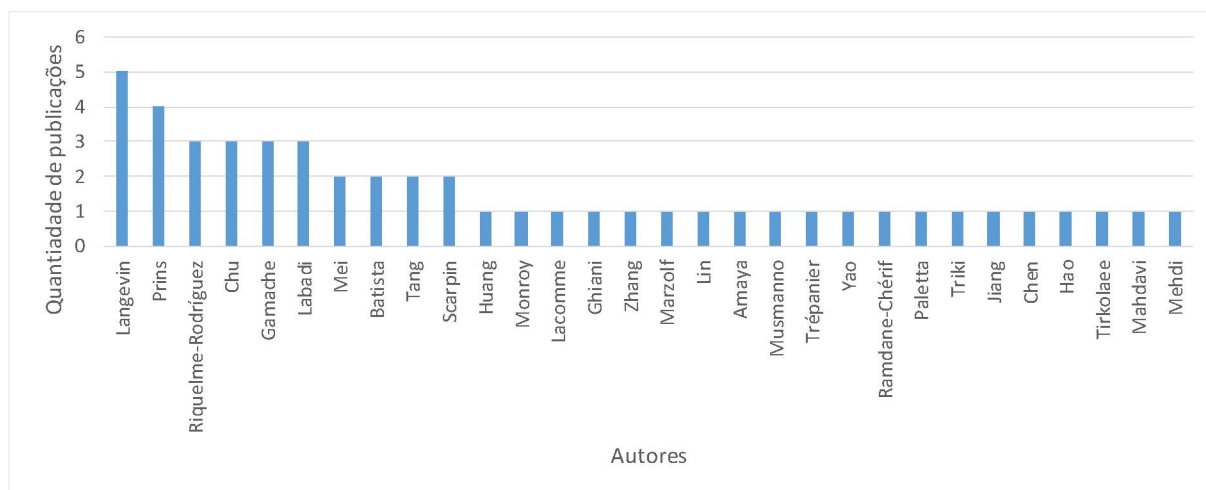
FONTE: O Autor (2018).

As análises a seguir consideraram apenas as publicações em periódicos. Isso porque os artigos publicados em eventos muitas vezes apresentam apenas resultados parciais que culminam em uma publicação dentro de periódicos. Os

assuntos abordados em eventos normalmente estão todos presentes nas revistas, exceto o trabalho de Kansou e Yassine (2009).

A análise do GRÁFICO 2.2 mostra todos os 30 autores que publicaram em revistas. Percebe-se que dois autores dominaram as publicações sobre o tema, são eles o André Langevin da Escola Politécnica de Montréal no Canadá que pertencente ao grupo CIRRELT (*Centre de recherche sur les réseaux d'entreprises*) e o Christian Prins da Universidade de Tecnologia de Troyes na França.

GRÁFICO 2.2 - PUBLICAÇÕES POR AUTOR



FONTE: O Autor (2018).

As revistas que mais contém publicações são a *Computers and Operations Research* e a *European Journal of Operational Research*, como pode ser analisado no GRÁFICO 2.3.

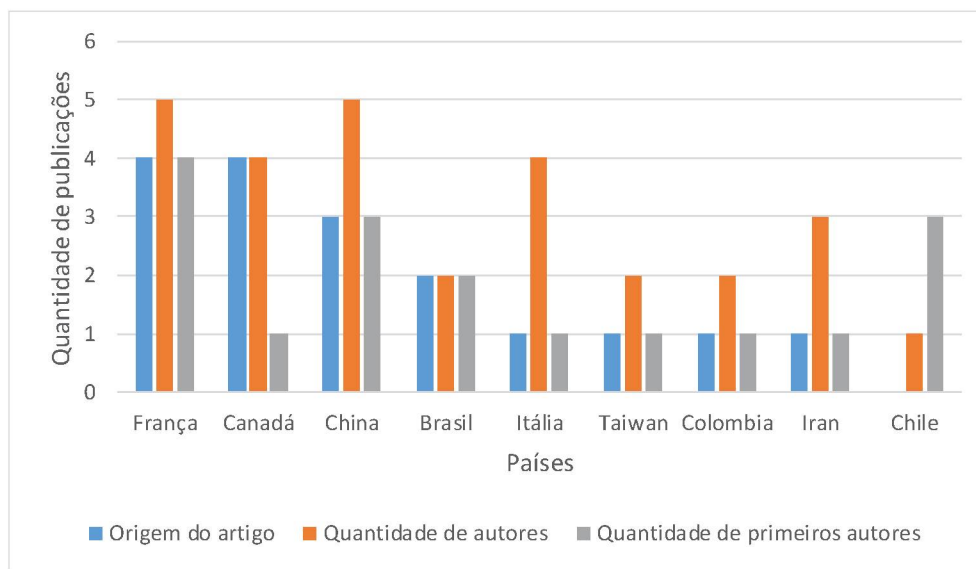
GRÁFICO 2.3 - REVISTAS QUE MAIS PUBLICARAM SOBRE O PCARP



FONTE: O Autor (2018).

Os países que agregam maior quantidade de publicações são a França e o Canadá, que produziram no total 8 dos artigos. Já os países que contam com mais autores sobre o assunto estão localizados na França e na China, sendo 10 no total. Essa análise pode ser feita por meio do GRÁFICO 2.4. É possível ver que a França também domina quando analisada a origem do primeiro autor.

GRÁFICO 2.4 - RELAÇÃO DE PUBLICAÇÃO POR PAÍSES



FONTE: O Autor (2018).

Esses dados mostram que é um assunto recente que começou a ser estudado em 2002. O volume de publicações não chega ser tão alto quanto em outros problemas de roteirização, mesmo assim tem sido abordado recorrentemente nos últimos anos. Com toda a gama de características apresentadas é possível notar que ainda a campo para o desenvolvimento científico acerca do tema.

3.3 MODELAGEM MATEMÁTICA PARA O PROBLEMA DE ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS COM MOVIMENTOS CONTÍNUOS

3.1 MODELAGEM MATEMÁTICA PRÉVIA

A modelagem a seguir foi apresentada na dissertação do autor (BATISTA, 2014), também publicada para mostrar aplicação a um problema real (BATISTA; SCARPIN, 2015) e usada para comparação com um modelo de serviço irregular (BATISTA; SCARPIN, 2014). Ela abrange o problema de inspeção e manutenção de ferrovias descrito no capítulo 1.

A formulação é baseada em um grafo não direcionado $G = (X, E)$ com n pontos, logo, $X = \{1, 2, \dots, n\}$ e m arestas com $E = \{1, 2, \dots, m\}$ que devem ser percorridos por nk carros definidos pelo conjunto $K = \{1, 2, \dots, nk\}$. Cada arco e é formado por um par de nós e é chamado de $x_{ij} = (x_i, x_j)$ ou $[i, j]$, estando associado a um custo c_{ij} . Para o roteamento, cada aresta é formada por um par de arcos direcionados que representam cada um dos sentidos possíveis de locomoção. A capacidade de cada carro é convertida na máxima distância que um carro pode percorrer em um dia, a qual neste caso é de uma aresta, fato que simplifica o modelo.

A periodicidade de cada arco é expressa na quantidade máxima de períodos consecutivos em que o arco deve ser atendido ao menos uma vez $MP(x_{ij})$. Por exemplo, se a periodicidade for de 15 dias, entre o dia 1 e o dia 15 deve ocorrer ao menos um atendimento, entre os dias 2 e 16 também e assim por diante. Tem-se que o horizonte de tempo H , é formado por np períodos $H = \{1, 2, \dots, np\}$, no qual cada período é simbolizado pela letra p . O modelo é baseado em Programação Linear Binária, em que as variáveis são:

$$\begin{aligned}
 x_{ijkp} & \begin{cases} 1, \text{ se o carro } k \text{ se desloca do ponto } i \text{ para o ponto } j \text{ no período } p \\ 0, \text{ caso contrário} \end{cases} \\
 pn_{ijp} & \begin{cases} 1, \text{ se o arco } x_{ij} \text{ não respeita a periodicidade no período } p \\ 0, \text{ caso contrário} \end{cases} \\
 f_{ikp} & \begin{cases} 1, \text{ se o carro } k \text{ fica parado no ponto } i \text{ no dia } p \\ 0, \text{ caso contrário} \end{cases}
 \end{aligned}$$

Os índices i e j estão associados a cada um dos arcos $[i, j]$, podendo ser eles os seus pontos finais ou iniciais. O índice k está associado aos carros e o índice p aos períodos. As variáveis pn_{ijp} e f_{ikp} foram criadas para possibilitar a factibilidade do problema, mesmo que o veículo fique parado em um mesmo ponto durante todo o horizonte de tempo. Cada variável pn_{ijp} tem associado a ela um custo PU_{ij} que é uma punição, penalidade, caso a periodicidade de um arco não seja atendida. Isso permite que haja um atraso no atendimento de alguns arcos, salientando que essa quantidade de atrasos é objetivo prioritário a ser minimizado. O uso de punições também é feito em Eglese (1994) e mais recentemente em Riquelme-Rodríguez, Gamache, Langevin (2014a, 2014b, 2016), quando alguma demanda não é atendida a tempo e a penalidade começa a interferir na função objetivo piorando-a.

A variável f_{ikp} permite que o carro folgue, o que é equivalente a ficar parado em algum ponto em um determinado dia, pois algumas vezes é possível obter uma solução com menos deslocamentos. Isso pode implicar que quando essas variáveis assumirem valor 1, muito provavelmente as periodicidades poderiam ser atendidas em menor tempo, ou seja, o horizonte de tempo pode reduzir.

A função objetivo é dada pela equação (1):

$$\min Z = \sum_{[i,j] \in E} \sum_{k=1}^{nk} \sum_{p=1}^{np} c_{ij} x_{ijkp} + \sum_{[i,j] \in E} \sum_{p=1}^{np} PU_{ij} pn_{ijp} \quad (1)$$

E o modelo está sujeito às seguintes restrições:

$$\sum_{[i,j] \in E} x_{ijkp} + f_{jkp} - \sum_{[i,j] \in E} x_{jik,p+1} - f_{jk,p+1} = 0 \quad \forall j \in X, \forall k \in K, \forall p \in H \quad (2)$$

$$\sum_{[i,j] \in E} x_{ijkp} + \sum_{[i,j] \in E} x_{jikp} + \sum_{i=1}^n f_{ikp} = 1 \quad \forall p \in H, \forall k \in K \quad (3)$$

$$\sum_{k=1}^{nk} \left(x_{ijkp} + x_{jikp} + x_{ijk,p+1} + x_{jik,p+1} + \dots \right. \\ \left. + x_{ijk,p+MP(x_{ij})-1} + x_{jik,p+MP(x_{ij})-1} \right) + pn_{ijp} \geq 1 \quad (4)$$

$$\forall [i, j] \in E, \forall p \in H$$

$$x_{ijkp}, pn_{ijp}, f_{ikp} \in \{0,1\} \quad \forall [i, j] \in E, \forall k \in K, \forall p \in H \quad (5)$$

A restrição (2) garante o fluxo diário dos carros, permitindo a folga em algum ponto j . Como serão formados ciclos, ao fim do período np o carro deve retornar para o mesmo local em que estava no dia 1. Os períodos que são computados com valores acima de np devem seguir uma lógica de transformação, sendo que o período $np + 1$ é igual ao dia 1 e assim por diante. Isso vale para todo o horizonte de tempo e para todas as restrições. Em (2) também se garante que o destino de um carro ao final do dia deve ser o ponto de partida no início do próximo dia, mesmo que não haja nenhum deslocamento e ele fique parado. Quando os carros estão em campo, podem ocorrer imprevistos e outros problemas que atrasem o atendimento à alguns arcos, se esse carro teve alguma folga planejada pelo modelo, essa folga pode compensar o tempo perdido.

A restrição (3) garante que todos os carros terão alguma designação para cada dia p , além disso, garante implicitamente que a capacidade de cada carro não seja excedida, visto que todas as variáveis são binárias e é possível no máximo atender um arco por dia. Tem-se basicamente que em cada dia, cada veículo deve executar um movimento, podendo deslocar-se de i para j , ou de j para i , ou simplesmente ficar parado em um ponto i .

Já a restrição (4) é a mais complexa. Ela se refere à periodicidade de cada trecho. Nela é possível que as variáveis pn_{ijp} assumam valor 1, ou seja, um arco pode ser atrasado em um dia ao custo de ser penalizado ao valor PU_{ij} na função objetivo. Em (4) cada periodicidade é analisada com a finalidade de verificar se a demanda nos intervalos de tempo é atendida. Cada intervalo de $MP(x_{ij})$ dias é avaliado de forma que se o arco é atravessado ao menos uma vez, pelo menos uma variável x_{ijkp} assumirá valor 1, se isso não ocorrer, a variável p_{ijp} obrigatoriamente assumirá esse valor.

Contudo, se a necessidade for de apenas uma passagem durante todo o horizonte de tempo, a restrição pode ser simplificada para apenas um dia p ($p = 1$, o que abrange todo o horizonte de tempo) e a variável pn_{ijp} pode ser descartada nessas ocasiões. O descarte dela se dá ao fato de não ser mais necessária visto que a passagem passa a se tornar obrigatória. Esse passo foi aplicado em todos os testes, obrigando ao menos um atendimento nos arcos x_{ij} com $MP(x_{ij}) = np$. A restrição (6) representa essa situação para todo arco que necessita de apenas uma passagem durante todo o horizonte de tempo. É apenas uma simplificação com a

finalidade de reduzir a quantidade de restrições a serem processadas para resolução do problema.

Supondo que o conjunto R contém todos os arcos com $MP(x_{ij}) = np$, a restrição que obriga o arco a ser atendido assume a seguinte forma:

$$\sum_{k=1}^{nk} \left(x_{ijk1} + x_{jik1} + x_{ijk2} + x_{jik2} + \dots + x_{ijk,MP(x_{ij})} + x_{jik,MP(x_{ij})} \right) \geq 1 \quad \forall [i,j] \in R \quad (6)$$

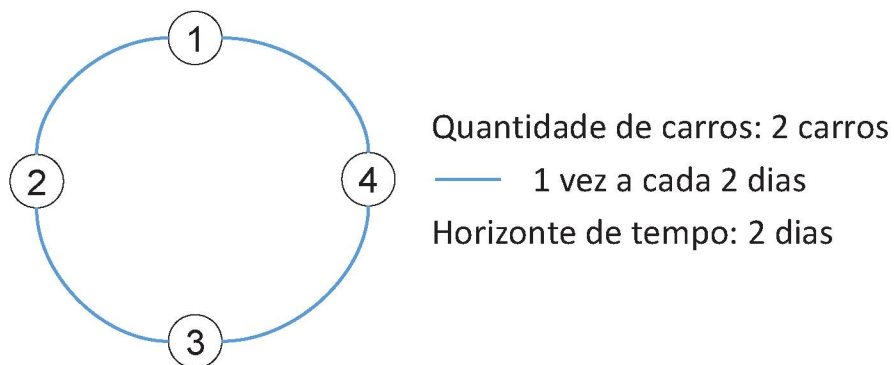
Em Lacomme, Prins e Ramdane-Chérif (2002a), o horizonte de tempo era muito curto, sendo considerado vantajoso utilizar uma metodologia designando uma combinação de dias para cada tarefa a fim de resolver o problema. Apesar de ser considerado complicado trabalhar com restrições de espaço e de tempo, como foi feito no modelo apresentado, a realidade fica muito bem representada.

Finalmente, a equação (5) determina que todas as variáveis sejam binárias. O modelo foi validado em um conjunto de 10 pequenas instâncias nas quais eram conhecidas suas soluções ótimas e também aplicado a 23 novas instâncias originadas de Golden, Dearmon e Baker (1983).

3.2 CARACTERÍSTICAS NÃO ABORDADAS

O modelo apresentado na seção anterior dá soluções viáveis e muito boas para o problema, porém algumas ocasiões não são contempladas pelo modelo. O exemplo hipotético da FIGURA 3.1 apresenta algumas dessas situações.

FIGURA 3.1 - EXEMPLO HIPOTÉTICO DE PCARP-CM

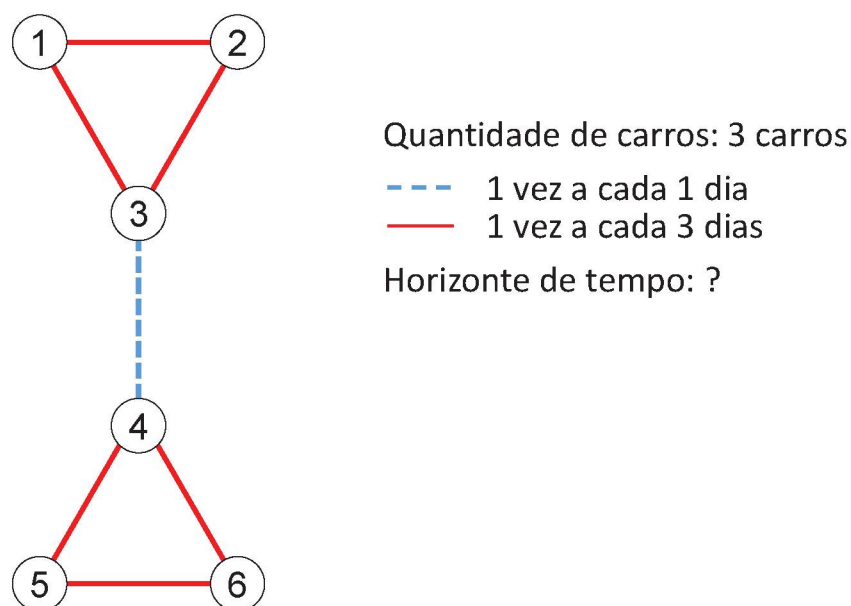


FONTE: O Autor (2018).

Considerado um horizonte de tempo de 2 dias para a resolução do problema, seria possível obter uma solução viável? Visto que os veículos são homogêneos e podem fazer o atendimento em qualquer arco dentro do horizonte de tempo, uma resposta lógica é fácil de ser encontrada. Basta possibilitar que as rotas de cada um dos dois veículos sejam finalizadas no que foi o ponto inicial de outro veículo, o carro 1 poderia fazer o movimento 1-2-3 e o carro 2 o movimento 3-4-1. Se a quantidade de veículos que sai no primeiro dia de algum ponto for igual a quantidade de veículos que retorna a esse ponto no último dia, então a solução se torna viável. Esse ajuste no modelo pode possibilitar soluções melhores do que quando os veículos estão restritos a terminar suas rotas em seus próprios pontos iniciais.

Outra alternativa seria aumentar o horizonte de tempo, no caso da Figura 3.1, ao supor um horizonte de tempo de 4 períodos, os veículos podem fazer as rotas 1-2-3-4-1 e 3-4-1-2-3. Porém, a determinação do tamanho do horizonte de tempo não é trivial. Como já mencionado na seção 2.2.1.2, o horizonte de tempo tem início e fim definidos para todos os modelos de PCARP e nessa variação não é diferente. A escolha do horizonte de tempo deve ser baseada nas periodicidades dos arcos, se o horizonte de tempo for equivalente à máxima periodicidade, soluções já podem ser encontradas. A extensão do horizonte de tempo pode acarretar em soluções cada vez melhores. Observe-se o problema da FIGURA 3.2:

FIGURA 3.2 - PROBLEMA DE DETERMINAÇÃO DO HORIZONTE DE TEMPO



FONTE: O Autor (2018).

Se considerado um horizonte de tempo de 3 dias, a melhor solução acarretará em um atraso do arco 3-4, porém é viável e a melhor para o modelo proposto. Um carro deve fazer o movimento 3-1-2-3, outro 3-4-3-3, ficando parado no ponto 3 no último dia, e outro 4-5-6-4, por exemplo. Porém, se explorado um horizonte de tempo de 6 dias as rotas: 3-4-3-4-3-4-3, 1-2-3-1-2-3-1 e 4-5-6-4-5-6-4 formam uma solução ótima sem atrasos.

Quando um problema tem apenas um carro e duas periodicidades diferentes parece lógico pensar que o horizonte de tempo pode ser o mínimo múltiplo comum dos valores das periodicidades. Porém, se analisada uma situação como a de quando o número de carros aumenta e o número de periodicidades também, o horizonte de tempo pode ser algum múltiplo do mínimo múltiplo comum, como no caso da FIGURA 3.2.

O horizonte de tempo deve então ser pré-definido por quem vai abordar o problema. A exemplo, uma situação contando com periodicidades que são formadas por números primos pode resultar em um mínimo múltiplo comum muito grande para compor horizonte de planejamento, tornando o problema inviável de ser resolvido. Outro problema poderia ser proposto tentando determinar o horizonte de tempo mínimo para PCARP-CM. Em alguns problemas como balanceamento das distribuições de tarefas em linhas de produção que operam ciclicamente, o tempo de ciclo é a variável a ser minimizada para melhorar a eficiência do regime permanente (LOPES; SIKORA; MAGATÃO, 2016).

3.3 PROPOSTA DE MELHORIA DO MODELO MATEMÁTICO

O modelo da seção 3.1 foi alterado para garantir a possibilidade de trocas de pontos iniciais ao final do horizonte de tempo para todos os carros. A formulação a seguir continua seguindo as mesmas ideias do modelo apresentado. Dado “um grafo não direcionado $G = (X, E)$ com n pontos, logo, $X = \{1, 2, \dots, n\}$, e m arestas com $E = \{1, 2, \dots, m\}$ que devem ser percorridos por nk carros definidos pelo conjunto $K = \{1, 2, \dots, nk\}$. Cada arco e é formado por um par de nós e é chamado de $x_{ij} = (x_i, x_j)$ ou $[i, j]$ e é associado a um custo c_{ij} ”.

Um resumo dos parâmetros de entrada que serão considerados no novo modelo pode ser encontrado no QUADRO 3.1:

QUADRO 3.1 - PARÂMETROS DE ENTRADA DO MODELO MATEMÁTICO

Parâmetro	Característica
$G = (X, E)$	Grafo não direcionado considerado
X	Conjunto de nós
E	Conjunto de arestas
$[i, j]$	Arco formado pelos pontos i e j
c_{ij}	Custo para atravessar ou servir o arco $[i, j]$
PU_{ij}	Custo da punição associada ao atraso do arco $[i, j]$
H	Conjunto contendo todos os períodos que compõe o horizonte de tempo
np	Número de períodos, $ H $
K	Conjunto de carros
nk	Número de carros
MP_{ij}	Periodicidade de $[i, j]$
S	Subconjunto das arestas $[i, j]$ com MP_{ij} menor que np
R	Subconjunto das arestas $[i, j]$ com MP_{ij} igual a np

FONTE: O Autor (2018).

Todas as variáveis continuam a ser usadas nesse modelo, porém é acrescentada uma variável inteira que irá contabilizar a quantidade de carros que sai e deve voltar para cada ponto no início e final do horizonte de tempo. As variáveis usadas são:

$$\begin{aligned}
 x_{ijkp} & \begin{cases} 1, \text{ se o carro } k \text{ se desloca do ponto } i \text{ para o ponto } j \text{ no período } p \\ 0, \text{ caso contrário} \end{cases} \\
 pn_{ijp} & \begin{cases} 1, \text{ se o arco } x_{ij} \text{ não respeita a periodicidade que se inicia no período } p \\ 0, \text{ caso contrário} \end{cases} \\
 f_{ikp} & \begin{cases} 1, \text{ se o carro } k \text{ fica parado no ponto } i \text{ no dia } p \\ 0, \text{ caso contrário} \end{cases} \\
 y_i & \text{ Contabiliza a quantidade de carros que sai do ponto } i \text{ no dia 1 e chega no ponto } i \text{ no dia } np
 \end{aligned}$$

As funções das variáveis p_{ijp} e f_{ikp} continuam as mesmas e os parâmetros podem ser lembrados em 3.1. A partir disso, o novo modelo é o seguinte:

$$\min Z = \sum_{[i,j] \in E} \sum_{k=1}^{nk} \sum_{p=1}^{np} c_{ij} x_{ijkp} + \sum_{[i,j] \in E} \sum_{p=1}^{np} PU_{ij} pn_{ijp} \quad (7)$$

Sujeito à:

$$\sum_{[i,j] \in E} x_{ijkp} + f_{jkp} = \sum_{[i,j] \in E} x_{jik,p+1} + f_{jk,p+1} \quad \forall j \in X, \forall k \in K, p = 1, \dots, np - 1 \quad (8)$$

$$\sum_{[i,j] \in E} x_{ijk1} + f_{ik1} = y_i \quad \forall i \quad (9)$$

$$\sum_{[i,j] \in E} x_{ijk,np} + f_{jk,np} = y_j \quad \forall j \quad (10)$$

$$\sum_{[i,j] \in E} x_{ijkp} + \sum_{[i,j] \in E} x_{jikp} + \sum_{i=1}^n f_{ikp} = 1 \quad \forall p \in H, \forall k \in K \quad (11)$$

$$\sum_{k=1}^{nk} \sum_{d=1}^{MP(i,j)} (x_{ijk,f(d+p-1)} + x_{jik,f(d+p-1)}) + pn_{ijp} \geq 1 \quad (12)$$

$$\forall [i,j] \in S, \forall p \in H; f(r) = r, \text{ se } r \leq np \text{ ou } f(r) = (r) \bmod (np), \text{ se } r > np$$

$$\sum_{k=1}^{nk} \sum_{p=1}^{np} (x_{ijkp} + x_{jikp}) \geq 1 \quad \forall [i,j] \in R \quad (13)$$

$$x_{ijkp}, pn_{ijp}, f_{ikp} \in \{0,1\} \quad \forall [i,j] \in E, \forall k \in K, \forall p \in H \quad (14)$$

$$y_i \in N^+$$

A função objetivo (7) permanece a mesma que (1). Agora, com a presença da variável y_i , é possível adicionar custos de incentivos aos carros para que eles partam de um determinado ponto, porém a complexidade do problema aumenta em virtude do novo objetivo. E se a determinação de pontos de partida estivesse associada a abertura de depósitos, o problema poderia se tornar um problema de localização de facilidades.

A equação (8) continua garantido o fluxo de veículos, porém agora não é necessário contabilizar o dia $np + 1$ como sendo o primeiro período. As equações (9) e (10) garantem que a quantidade de carros que sai de um ponto no dia 1, será igual a quantidade de carros que voltará a esse ponto no final do horizonte de tempo np . A equação (11) continua garantindo que a cada dia, cada carro terá uma tarefa, seja um deslocamento ou ficar parado.

A restrição (4) do modelo anterior se tornou as restrições (12) e (13), que resumem a simplificação de (6). A restrição (12) é a inequação que visa garantir os atendimentos de todos os arcos com periodicidade menor que o tamanho do horizonte de tempo, se algum desses arcos for atrasado ocorrerá uma punição na função objetivo. A ideia de conversão dos períodos com valor acima de np está

representada pela função $f(r)$, o período $np + 1$ se torna o período 1, $np + 2$ o período 2 e assim por diante. Essa conversão é feita com a função resto (mod), na qual os períodos acima de np se tornam o resto da divisão por np .

A restrição (13) garante que algum atendimento será feito em todos os arcos com periodicidade igual ao horizonte de tempo. Um atraso em algum desses arcos não pode ser permitido, pois significaria que eles nunca seriam atendidos. Em (14) as variáveis binárias e inteiras são estabelecidas, lembrando ainda que as variáveis inteiras devem ser positivas maiores iguais a zero.

3.4 IMPLEMENTAÇÃO E RESULTADOS PARCIAIS DO MODELO MELHORADO

A dissertação de Batista (2014) valida os resultados do modelo ao aplicá-lo em 10 pequenos problemas cujas soluções eram possíveis de serem encontradas de maneira intuitiva, ou ainda, o problema era criado partindo de uma solução. Essas 10 instâncias receberam o nome de *gvb* e seus respectivos resultados no trabalho foram os listados na TABELA 3.1, já os resultados com o novo modelo estão listados na TABELA 3.2.

TABELA 3.1 - RELAÇÃO DOS 10 PROBLEMAS GERADOS, PROPRIEDADES E RESULTADOS.

Problema	Nós	Arestas	Period.	Carros	Neces.	Dias	Solução	Atrasos	Folgas	Tempo (s)
<i>gvb1</i>	16	24	2	9	36	4	36*	0	0	2,40
<i>gvb2</i>	16	24	3	3	36	12	36*	0	0	364,85
<i>gvb3</i>	16	24	3	6	36	6	36*	0	0	338,99
<i>gvb4</i>	16	24	2	4	32	8	32*	0	0	69672,00
<i>gvb5</i>	6	7	2	3	18	6	18*	0	0	0,08
<i>gvb6</i>	6	7	2	1	8	8	8*	0	0	0,03
<i>gvb7</i>	12	24	2	7	42	6	42*	0	0	4,34
<i>gvb8</i>	8	12	2	2	16	8	16*	0	0	0,39
<i>gvb9</i>	14	23	3	10	60	6	60*	0	0	1,53
<i>gvb10</i>	11	12	2	1	16	24	24*	0	0	239,06

* Solução ótima

FONTE: Batista (2014).

A TABELA 3.1 traz um resumo das características de cada instância e seus resultados. A primeira coluna apresenta o nome, a segunda a quantidade de nós, a terceira a quantidade de arestas, a quarta o número de periodicidades diferentes, a quinta a quantidade de carros, a sexta as necessidades (soma total da quantidade mínima de atendimentos que devem ser feitos), a sétima o horizonte de tempo, a

oitava a solução, a nona a quantidade de atrasos na solução, a décima a quantidade de folgas em cada solução e a décima primeira o tempo de resolução em segundos.

Os problemas foram resolvidos pelo software CPLEX 12.4 em um computador Intel Core i7 com 2 Ghz de processamento e 4Gb de memória RAM e sistema operacional 64 bits. O novo modelo foi aplicado a mesma classe de instâncias, porém utilizando o software Gurobi 6.5. A TABELA 3.2 apresenta todos os resultados do novo modelo, que também chegou no ótimo no espaço de soluções maior com tempo aceitável.

TABELA 3.2 - RESULTADOS DOS 10 PROBLEMAS *gvb* PARA O MODELO 2018

Problema	Modelo 2014				Modelo 2018			
	Solução	Atrasos	Folgas	Tempo (s)	Solução	Atrasos	Folgas	Tempo (s)
<i>gvb1</i>	36*	0	0	2,40	36*	0	0	0,68
<i>gvb2</i>	36*	0	0	364,85	36*	0	0	278,27
<i>gvb3</i>	36*	0	0	338,99	36*	0	0	1093,85
<i>gvb4</i>	32*	0	0	69672,00	32	0	0	59682
<i>gvb5</i>	18*	0	0	0,08	18*	0	0	0,03
<i>gvb6</i>	8*	0	0	0,03	8*	0	0	0,1
<i>gvb7</i>	42*	0	0	4,34	42*	0	0	1,07
<i>gvb8</i>	16*	0	0	0,39	16*	0	0	0,02
<i>gvb9</i>	60*	0	0	1,53	60*	0	0	8,02
<i>gvb10</i>	24*	0	0	239,06	24*	0	0	407,03

* Solução ótima

FONTE: O Autor (2018).

Outras 23 instâncias foram criadas a partir de Golden, Dearmon e Baker (1983) no trabalho de Batista (2014) e se assemelham às operações já feitas em outros trabalhos (CHU; LABADI; PRINS, 2004c, 2005, 2006; LACOMME; PRINS; RAMDANE-CHÉRIF, 2005). O processo foi descrito da seguinte forma:

“... as novas adaptações que foram feitas nas instâncias são as seguintes:

- Foram mantidos apenas os custos e as ligações entre os nós e os grafos foram considerados não direcionados;
- O número de periodicidades variaram entre 2, 3 ou 4;
- O número de carros que trabalham em cada grafo foi definido aleatoriamente entre 1 a 6;
- A necessidade de cada arco foi definida aleatoriamente, se um grafo tem 3 periodicidades, foram definidos arcos que necessitavam de 3 passagens, 2 passagens e 1 passagem ao longo do horizonte de tempo e assim sucessivamente;
- O número de necessidades de cada problema é a soma das necessidades de cada arco (número de atendimentos no arco);
- O horizonte de tempo é dado pelo número de necessidades do problema multiplicado por 1,2 arredondado para cima, tudo dividido pelo número de carros e também arredondado para cima;

- A periodicidade de cada arco é dado pelo horizonte de tempo, dividido por sua necessidade e arredondado para cima;
- Logo, a necessidade de cada arco também pode ser dada pela periodicidade de cada arco dividido pelo horizonte de tempo e arredondado para cima. Por exemplo, se a periodicidade de um arco é de 11 dias e o horizonte de tempo é de 20 dias, a necessidade desse arco é de 2 passadas durante o horizonte de tempo.” (BATISTA, 2014, p. 59)

Essas instâncias aqui receberão o nome de *pgdbcm*, por serem uma versão periódica de *gdb* com movimentos contínuos. Depois de criados, esses problemas também foram resolvidos utilizando o software CPLEX 12.4, porém em computadores com processador Intel Core i5 com processamento de 3,1 GHz com 8 Gb de memória RAM em um sistema operacional de 64 bits. Os trabalhos foram limitados a uma resolução de 24 horas, ou 86400 segundos, e os resultados encontrados, bem como as características, estão na TABELA 3.3.

TABELA 3.3 - RESULTADOS DAS 23 INSTÂNCIAS *pgdbcm*.

Problema	Nós	Arestas	Period.	Carros	Neces.	Dias	Solução	Atrasos	Folgas	Tempo (s)
<i>pgdbcm1</i>	12	22	2	1	32	38	4450	4	0	39000 ¹
<i>pgdbcm2</i>	12	26	3	1	44	53	11622	11	1	70423 ¹
<i>pgdbcm3</i>	12	22	2	2	31	19	340	0	3	36906 ¹
<i>pgdbcm4</i>	11	19	3	2	34	21	3555	3	1	49595 ¹
<i>pgdbcm5</i>	13	26	4	2	53	32	1750	1	1	86400 ²
<i>pgdbcm6</i>	12	22	2	3	34	14	456	0	3	86400 ²
<i>pgdbcm7</i>	12	22	3	3	37	45	518	0	5	55767 ¹
<i>pgdbcm8</i>	27	46	4	3	67	27	12395	12	0	86400 ²
<i>pgdbcm9</i>	27	51	3	4	78	24	13454	13	2	86400 ²
<i>pgdbcm10</i>	12	25	4	4	70	21	734	0	6	86400 ²
<i>pgdbcm11</i>	22	45	2	1	50	60	64437	64	2	31510 ¹
<i>pgdbcm12</i>	13	23	3	1	33	40	536	0	0	86400 ²
<i>pgdbcm13</i>	10	28	2	2	33	20	620	0	5	86400 ²
<i>pgdbcm14</i>	7	21	3	2	32	20	145*	0	7	4,21
<i>pgdbcm15</i>	7	21	4	2	48	29	140	0	7	63135 ¹
<i>pgdbcm16</i>	8	28	2	3	40	24	182	0	28	35301 ¹
<i>pgdbcm17</i>	8	28	3	3	52	21	168	0	4	77693 ¹
<i>pgdbcm18</i>	9	36	4	3	81	33	415	0	6	86400 ²
<i>pgdbcm19</i>	8	11	3	4	29	12	125	0	17	47566 ¹
<i>pgdbcm20</i>	11	22	4	4	53	16	2280	2	1	67903 ¹
<i>pgdbcm21</i>	11	33	2	5	46	12	212	0	10	38181 ¹
<i>pgdbcm22</i>	11	44	4	5	88	22	433	0	17	86400 ²
<i>pgdbcm23</i>	11	55	2	6	82	17	336	0	15	86400 ²

¹ Problemas que esgotaram a memória do computador antes de chegar a uma solução ótima ou atingir o tempo limite

² Problemas que rodaram até o tempo limite estabelecido

* Solução ótima

FONTE: Batista (2014).

As mesmas condições de limitação de tempo foram aplicadas para a utilização do novo modelo, porém todos os resultados obtidos a partir desse momento foram obtidas utilizando o software Gurobi 8.0.1 em notebooks com sistema operacional 64 bits com processador Intel Core i7-5500U com 2.4Ghz de processamento e 8 Gb de memória RAM salvo quando indicado. O desenvolvimento ocorreu no ambiente de programação VisualBasic.NET utilizando a biblioteca disponibilizada pelo *solver*. Os resultados alcançados, seus respectivos *lower bounds* e gaps em comparação com o modelo de 2014 estão na TABELA 3.4.

TABELA 3.4 - RESULTADO *pgdbcm* COMPARANDO MODELOS

Problema	Modelo 2014				Modelo 2018					
	Bound	Solução	Gap	Tempo (s)	Bound	Solução	Gap	Atrasos	Folgas	Tempo (s)
<i>pgdbcm1</i>	378	4450	91,51%	39000 ¹	451	451*	0,00%	0	0	29869
<i>pgdbcm2</i>	510	11622	95,61%	70423 ¹	518	614	15,64%	0	1	86400 ²
<i>pgdbcm3</i>	315	340	7,35%	36906 ¹	315	340	7,35%	0	3	86401 ²
<i>pgdbcm4</i>	447,79	3555	87,40%	49595 ¹	449	2519	82,18%	2	1	86401 ²
<i>pgdbcm5</i>	659	1750	62,34%	86400 ²	659	6737	90,22%	6	1	86400 ²
<i>pgdbcm6</i>	418	456	8,33%	86400 ²	418	456	8,33%	0	3	86401 ²
<i>pgdbcm7</i>	491	518	5,21%	55767 ¹	494	508	2,76%	0	4	86400 ²
<i>pgdbcm8</i>	325	12395	97,38%	86400 ²	325	8395	96,13%	8	0	86400 ²
<i>pgdbcm9</i>	439	13454	96,74%	86400 ²	439	44425	99,01%	44	2	86401 ²
<i>pgdbcm10</i>	684	734	6,81%	86400 ²	684	766	10,70%	0	4	86400 ²
<i>pgdbcm11</i>	411	64437	99,36%	31510 ¹	416	442	5,88%	0	2	86400 ²
<i>pgdbcm12</i>	466,45	536	12,98%	86400 ²	484	536	9,70%	0	0	86400 ²
<i>pgdbcm13</i>	616	620	0,65%	86400 ²	620	620*	0,00%	0	5	10
<i>pgdbcm14</i>	145	145*	0,00%	4,21	145	145*	0,00%	0	7	33
<i>pgdbcm15</i>	137	140	2,14%	63135 ¹	137	140	2,14%	0	7	86400 ²
<i>pgdbcm16</i>	176	182	3,30%	35301 ¹	176	182	3,30%	0	28	86402 ²
<i>pgdbcm17</i>	161	168	4,17%	77693 ¹	161	168	4,17%	0	5	86400 ²
<i>pgdbcm18</i>	363	415	12,53%	86400 ²	369	379	2,64%	0	14	86400 ²
<i>pgdbcm19</i>	121	125	3,20%	47566 ¹	125	125*	0,00%	0	47	109
<i>pgdbcm20</i>	231,66	2280	89,84%	67903 ¹	240	2262	89,39%	2	1	86400 ²
<i>pgdbcm21</i>	204	212	3,77%	38181 ¹	210	212	0,94%	0	10	86402 ²
<i>pgdbcm22</i>	420,75	433	2,83%	86400 ²	423	435	2,76%	0	13	86400 ²
<i>pgdbcm23</i>	329	336	2,08%	86400 ²	329	336	2,08%	0	14	86402 ²

¹ Problemas que esgotaram a memória do computador antes de chegar a uma solução ótima ou atingir o tempo limite

² Problemas que rodaram até o tempo limite estabelecido

* Solução ótima

FONTE: O Autor (2018).

A análise dos resultados não tem como objetivo fazer a comparação entre softwares, visto que com o decorrer do tempo novas versões são lançadas com grandes melhorias. Nota-se que a gestão de memória do Gurobi é um ponto

fortíssimo do software, que mesmo em versões anteriores já não a esgotava, permitindo atingir valores mais baixos da função objetivo para o PCARP-CM. Tanto CPLEX quanto Gurobi são softwares de referência no mercado e são amplamente utilizados por diversos pesquisadores. Aqui optou-se por trabalhar com Gurobi pelo seu bom desempenho, sendo que nos testes foi utilizada sua configuração *default*.

É possível perceber que o modelo anterior trouxe melhores resultados em 4 instâncias, houve empate em 10 instâncias e o modelo 2018 foi superior em 9. Os *Lowers bounds* apresentados são os oferecidos pelo software após resolução. Como os modelos são distintos e o espaço solução do novo modelo é maior, o resultado encontrado para o primeiro modelo serve como resposta para o novo, mas o contrário muitas vezes não é válido.

Além disso, a combinação do novo modelo com o *solver* Gurobi atingiu melhores *bounds* que, no caso de problemas de minimização, são *lower bounds*. O valor ótimo de um problema só pode ser provado quando a solução encontrada for igual ao valor mínimo ou máximo esperado do problema (*bound*), *lower bound* e *upper bound* respectivamente. Com o novo modelo foi possível comprovar 4 ótimos globais.

Os melhores resultados encontrados com o primeiro modelo são justificados pela busca no espaço solução menor. Caso houvesse um maior tempo de resolução ou uma alternativa de aceleração no processo de busca por soluções, o novo modelo seria capaz de atingir no mínimo os mesmos resultados. O fato de permitir a troca dos pontos de partida ao final do último período aumentam em muito o espaço solução, pois ao escolher um ponto inicial não obrigatoriamente já é conhecido o retorno.

Para diversificar mais os testes e avaliar as condições do horizonte de tempo utilizando mínimos múltiplos comuns, modificou-se algumas das 23 instâncias para atender alguns critérios. As periodicidades envolvidas, bem como o horizonte de tempo, foram aproximadas de outros valores que culminassem em um mínimo múltiplo comum viável para cada problema. As mudanças feitas geraram 16 novos problemas, que estão apresentadas na TABELA 3.5. Algumas instâncias não foram modificadas por já trabalharem com mínimos múltiplos comuns. O número de arcos, nós, carros, bem como a estrutura do grafo foi mantida. Quando alterado os valores das periodicidades, procurou-se deixar também o número de necessidades próximo do que já vinha sendo trabalhado.

TABELA 3.5 - NOVAS 23 INSTÂNCIAS *pgdbcm*

Instância Geradora	Periodicidades	Neces.	Nova instância gerada	Novas periodicidades	Neces.
<i>pgdbcm1</i>	19-38	32	-	-	-
<i>pgdbcm2</i>	18-27-53	44	<i>pgdbcm24</i>	18-27-54	44
<i>pgdbcm3</i>	10-19	31	<i>pgdbcm25</i>	10-20	31
<i>pgdbcm4</i>	7-11-21	34	<i>pgdbcm26</i>	8-12-24	34
<i>pgdbcm5</i>	8-11-16-32	53	<i>pgdbcm27</i>	8-16-32	50
<i>pgdbcm6</i>	7-14	34	-	-	-
<i>pgdbcm7</i>	6-8-16	37	<i>pgdbcm28</i>	8-16	33
<i>pgdbcm8</i>	7-9-14-27	67	<i>pgdbcm29</i>	10-15-30	63
<i>pgdbcm9</i>	7-12-24	78	<i>pgdbcm30</i>	8-12-24	78
<i>pgdbcm10</i>	6-7-12-21	70	<i>pgdbcm31</i>	6-8-12-24	70
<i>pgdbcm11</i>	30-60	50	-	-	-
<i>pgdbcm12</i>	14-20-40	33	<i>pgdbcm32</i>	14-21-42	33
<i>pgdbcm13</i>	10-20	33	-	-	-
<i>pgdbcm14</i>	7-10-20	32	<i>pgdbcm33</i>	5-10-20	36
<i>pgdbcm15</i>	8-15-29	48	<i>pgdbcm34</i>	10-15-30	42
<i>pgdbcm16</i>	12-24	40	-	-	-
<i>pgdbcm17</i>	7-11-21	52	<i>pgdbcm35</i>	8-12-24	54
<i>pgdbcm18</i>	9-11-17-33	81	<i>pgdbcm36</i>	9-12-18-36	81
<i>pgdbcm19</i>	4-6-12	29	-	-	-
<i>pgdbcm20</i>	4-6-8-16	53	<i>pgdbcm37</i>	4-8-16	53
<i>pgdbcm21</i>	6-12	46	-	-	-
<i>pgdbcm22</i>	6-8-11-22	88	<i>pgdbcm38</i>	6-8-12-24	88
<i>pgdbcm23</i>	9-17	82	<i>pgdbcm39</i>	9-18	82

FONTE: O Autor (2018).

Os novos 16 problemas também foram modelados e resolvidos pelo Gurobi 8.0.1 usando a mesma configuração de notebook e os resultados são apresentados na TABELA 3.6. Como esperado, é possível perceber que os valores alcançados se assemelham aos resultados das instâncias geradoras devido ao porte dos problemas, porém são diferentes mostrando a diversificação. É importante ressaltar que o tempo computacional de resolução de um problema não é afetado apenas pelo tamanho da rede, mas também pela geometria do grafo (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014b).

Quanto tempo seria necessário para alcançar a solução ótima? A solução encontrada é a melhor possível e basta conseguir prová-la? São perguntas difíceis de serem respondidas. O estudo a ser desenvolvido a seguir visa mostrar alternativas para a resolução do PCARP-CM. A próxima seção apresentará um Algoritmo de Otimização de Colônia de Formigas (ACO) usando conceitos de busca tabu granular e busca local. Como visto aqui, o espaço solução interfere no

resultado alcançado pelo *solver*, de modo que um algoritmo baseado na busca de soluções dentro de espaços restritos, será proposto na sequência. O nome do algoritmo é *Iterative Restricted Space Search* (IRSS), Busca Iterativa em Espaços Restritos, e foi proposto em Pécora Jr. (2008).

TABELA 3.6 - RESULTADOS DAS NOVAS INSTÂNCIAS PGDCM

Problema	Modelo 2018					
	Bound	Solução	Gap	Atrasos	Folgas	Tempo (s)
<i>pgdbcm24</i>	517	582	11,17%	0	4	86400
<i>pgdbcm25</i>	323	340	5,00%	0	5	86400
<i>pgdbcm26</i>	447	508	12,01%	0	8	86400
<i>pgdbcm27</i>	639	676	5,47%	0	8	86400
<i>pgdbcm28</i>	402	444	9,46%	0	9	86405
<i>pgdbcm29</i>	305	339	10,03%	0	17	86400
<i>pgdbcm30</i>	375	21449	98,25%	21	2	86401
<i>pgdbcm31</i>	684	748	8,56%	0	16	86400
<i>pgdbcm32</i>	523	523*	0,00%	0	1	73300
<i>pgdbcm33</i>	153	157	2,55%	0	2	86400
<i>pgdbcm34</i>	120	120*	0,00%	0	15	57310
<i>pgdbcm35</i>	167	170	1,76%	0	15	86400
<i>pgdbcm36</i>	369	381	3,15%	0	20	86400
<i>pgdbcm37</i>	240	264	9,09%	0	3	86400
<i>pgdbcm38</i>	423	426	0,70%	0	29	86400
<i>pgdbcm39</i>	329	336	2,08%	0	20	86403

* Solução ótima

FONTE: O Autor (2018).

4 O ALGORITMO DE OTIMIZAÇÃO DA COLÔNIA DE FORMIGAS

A resolução de problemas aplicados em contextos reais como o PCARP-CM não é uma tarefa fácil, mesmo para *solvers* comerciais como o Gurobi ou CPLEX. Os problemas classificados como PCARP usualmente tem ótimos provados para uma quantidade pequena e limitada de instâncias (CHU; LABADI; PRINS, 2004c; MONROY; AMAYA; LANGEVIN, 2013). Alternativamente o problema é abordado por heurísticas (CHU; LABADI; PRINS, 2004a), Algoritmos Meméticos (LACOMME; PRINS; RAMDANE-CHÉRIF, 2005; MEI *et al.*, 2011; ZHANG *et al.*, 2016), Adaptative Large Neighborhood Search (RIQUELME-RODRÍGUEZ; GAMACHE; LANGEVIN, 2014a) e Otimização de Colônia de Formigas, Ant Colony Optimization (ACO) (HUANG; LIN, 2014).

O ACO tem alcançado bons resultados para problemas de roteamento dinâmico, caso em que os custos podem ser modificados após algum movimento (DORIGO; STÜTZLE, 2010). O PCARP-CM tem uma semelhança com essa característica, pois após cada movimento o prazo para atender cada arco fica mais próximo e caso algum atrase influenciará muito na função objetivo. Baseado nisso optou-se por implementar um ACO, o qual alcançou resultados competitivos.

O algoritmo ACO é inspirado pelo comportamento de formigas procurando comida e foi proposto em Coloni, Dorigo e Maniezzo (1991). As formigas espalham feromônios durante essa procura, isso faz com que seja criado um rastro que influencia a escolha dos caminhos das próximas formigas que saírem em busca de alimento. No algoritmo, uma informação numérica representa os rastros de feromônio, guardando a experiência adquirida com as buscas. Toda informação disponível é usada para gerar um movimento randômico para cada formiga e construir uma solução.

Algoritmos construtivos geram soluções para problemas adicionando componentes em uma solução até que ela seja completa. Normalmente não tem como retroceder após uma escolha nesse tipo de processo, por exemplo, em Heurísticas de Construção Gulosa (*Greedy Construction Heuristics*) o componente que traz o máximo benefício míope para o problema é adicionado a solução. Usualmente, esse tipo de algoritmo gera soluções com qualidade melhor do que soluções aleatórias, mas como desvantagem gera um número limitado de soluções.

Além disso, a escolha precoce de alguns componentes pode prejudicar a solução com movimentos pobres perto da fase final do algoritmo.

Após a obtenção de uma solução completa é possível a aplicação de algoritmos de busca local, também conhecidos como melhoria local. Esse tipo de algoritmo melhora a solução iterativamente buscando em uma vizinhança previamente limitada. Quando encontrada uma solução melhor, ela deve substituir a solução corrente e o processo de busca continua, agora dentro da vizinhança da nova solução. A escolha de quando substituir uma solução é dada por uma regra de pivoteamento muito importante nesse processo: é possível substituir a solução após olhar toda a vizinhança e escolher a melhor (regra da melhor melhoria) ou substituí-la logo após encontrar uma solução melhor (regra da primeira melhoria). Vale ressaltar que a escolha de uma boa vizinhança para exploração é essencial para a performance do algoritmo.

O ACO constrói soluções estocásticas permitindo a abrangência de uma grande variedade de diferentes soluções. O processo de construção adiciona cada componente, um por um, considerando a experiência de outras formigas e os dados disponíveis no momento, como deslocamento e prazos de atendimento. Quando se trata de ACO, sempre que é possível criar um procedimento que gere soluções para um problema de otimização discreta, então, em princípio, é possível aplicar o ACO (DORIGO; STÜTZLE, 2010).

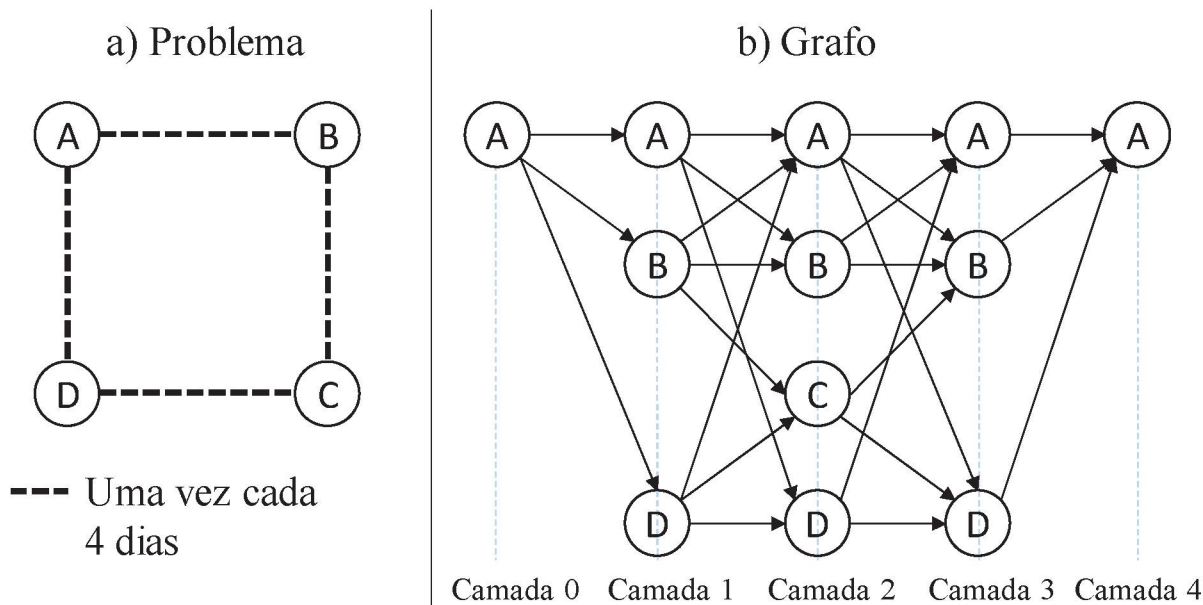
4.1 A REPRESENTAÇÃO DO PROBLEMA

Gerar soluções para o PCARP-CM não é uma tarefa trivial devido a continuidade dos movimentos e da falta de um depósito como referência. Além disso, tem-se a ciclicidade do problema, o que aumenta o número de soluções simétricas. Uma solução pode ser representada por uma sequência de cidades visitadas: tendo-se a cidade de partida é necessário saber aonde o carro termina em cada dia, o que é o mesmo que saber a cidade de partida do próximo dia. Soluções com a mesma sequência de visitas independentemente do dia ou do sentido são soluções simétricas e acarretam no mesmo valor da função objetivo.

O processo construtivo aqui desenvolvido consiste na definição de uma rota em um grafo direcionado. Primeiramente, é selecionado um vértice inicial para ser o ponto de partida, sendo que quanto mais conexões ele tiver maior sua probabilidade

de ser sorteado. Um vértice inicial com mais adjacências pode conduzir a soluções mais variadas que serão usadas posteriormente em uma hibridização. A partir desse ponto inicial um grafo é gerado para cada carro e é baseado em camadas; a camada inicial contém apenas o vértice selecionado (inicial) e a camada seguinte inclui apenas vértices adjacentes ao vértice inicial e ele mesmo com todas as possíveis conexões. O procedimento segue de camada para camada contendo todos os possíveis movimentos, ligando os vértices possíveis, até chegar a camada final. Nem todos os nós adjacentes são possíveis conexões em todas as camadas porque a última deve conter apenas os possíveis vértices finais. Por exemplo, se apenas um carro está disponível no problema, o último vértice a ser visitado na última camada é obrigatoriamente o vértice inicial. Quando se tem mais carros é necessário avaliar todas as possibilidades baseado na distância em períodos até chegar nos possíveis pontos finais. Com mais de um carro, o grafo do segundo carro só é criado após a formiga completar os movimentos do primeiro carro e o processo assim segue. Existem $np + 1$ camadas para cada carro, a FIGURA 4.1 representa um exemplo simples de problema e a sua representação do grafo.

FIGURA 4.1 - UM EXEMPLO DE PROBLEMA (a) E SUA REPRESENTAÇÃO NO GRAFO (b).



FONTE: o autor (2018)

Deixar uma camada e ir até a próxima representa um movimento feito por um carro. O grafo pode dar a falsa sensação de um simples problema, porém após cada movimento, as demandas mudam dinamicamente podendo alterar custos. Em

problemas periódicos, a demanda sofre influência do período e da rota do carro. No ACO proposto, as formigas começam no ponto inicial e probabilisticamente vão se movendo até formar uma rota completa para todos os carros, ou seja, se movimentam através das camadas.

A definição de um ponto inicial ajuda na busca de soluções, evitando muitas soluções simétricas. Para aumentar a efetividade do ACO, o algoritmo sofreu uma hibridização com alguns conceitos de Busca Tabu Granular, *Granular Tabu Search* (GTS) (TOTH;VIGO, 2003). O GTS é aplicado na otimização de problemas que envolvem grafos usando o que é chamado de vizinhança granular nos algoritmos de busca local para reduzir o tempo computacional. Em cada iteração a busca passa a ser mais orientada porque ela ocorre entre vizinhos com uma característica desejável. Por exemplo, para o Problema de Roteamento de Veículos, um grafo esparsos pode ser criado para guiar a busca e ele vai conter apenas arcos relevantes, por exemplo, com distâncias curtas.

O algoritmo ACO aqui proposto conterá dois grafos esparsos que vão proibir alguns movimentos. O primeiro grafo será usado para proibir os movimentos que mantêm o carro no mesmo lugar. O segundo será modificado a cada movimento, eliminando algumas possibilidades que seguem. Primeiramente, se uma formiga atravessa um arco que tem periodicidade igual ao horizonte de tempo não será necessário atravessá-lo novamente. Outra situação pode ocorrer após atender outros tipos de arcos, pois após um atendimento em um arco é razoável não o atender novamente tão cedo, então ele será proibido de ser atravessado alguns períodos para frente e outros para trás. A quantidade de períodos bloqueados para frente e para trás é do tamanho da periodicidade dividido por 1,5, valor prático. O ideal seria atravessar exatamente ao final do prazo, porém isso seria uma imposição muito forte para o modelo e não ajudaria a resolver o problema. Esse processo pode vir a bloquear todos os movimentos em algum ponto e, caso isso aconteça, primeiro é desconsiderado apenas os movimentos do segundo grafo esparsos e se o problema persistir, os movimentos do primeiro grafo também são liberados.

4.2 O MOVIMENTO DAS FORMIGAS

Todas as formigas iniciam pelo nó inicial na primeira camada e seguem em frente até gerar uma solução completa. Cada movimento representa uma variável de

decisão do problema, por exemplo, o movimento de um carro de um nó para outro em um dado dia. O processo de decisão considera o status atual de todos os arcos e os rastros de feromônio deixado por formigas anteriores; cada opção de movimento tem uma probabilidade que utiliza essas informações. Iterativamente, a adição de cada movimento compõe uma solução.

Algumas possibilidades podem ser mais atrativas que outras no processo de decisão. Por exemplo, se ao final do 7º dia de planejamento uma formiga encontra-se em uma cidade A e precisa decidir entre ir para uma cidade C ou B. Neste caso, ela tem as seguintes informações à disposição: o arco A-C foi atravessado 6 dias atrás e tem uma periodicidade de 30 dias e o arco A-B ainda não recebeu o atendimento que deve ocorrer a cada 15 dias. Baseado nessa informação parece mais promissor escolher visitar a cidade B, então é possível interferir na decisão aleatória da formiga dando mais probabilidade para a escolha da cidade B como a próxima visita.

Um fator de visibilidade η_{ij} é calculado para cada opção de movimento no momento, de maneira que é feita uma análise dos movimentos passados, isso é chamado de informação heurística. Essa informação quantifica a qualidade da escolha do arco e irá induzir que o movimento avaliado está zerando a quantidade de dias sem atendimento enquanto os outros arcos estarão um dia mais próximo do prazo final. Considerando a variável $last_{ij}$ o último dia em que uma aresta (i, j) foi atendida, d o dia em que o movimento avaliado está acontecendo e $g(ij)$ uma função auxiliar que ajudará no cálculo, o fator de visibilidade pode ser calculado como:

$$g(ij) = MP(i, j) + \sum_{[k, l] \in E/[i, j]} (last_{kl} + MP(k, l) - d) \quad \forall [i, j] \in E \quad (15)$$

$$\eta_{ij} = \begin{cases} g(ij) & \text{se } g(ij) > 0 \\ 1 & \text{se } g(ij) = 0 \\ -g(ij)^{-1} & \text{se } g(ij) < 0 \end{cases} \quad \forall [i, j] \in E \quad (16)$$

Quanto maior o valor de $g(ij)$ na equação (15), mais atrativo é fazer o movimento de i para j e valores negativos indicam que alguma periodicidade (dada por MP) foi violada. A visibilidade η_{ij} necessita ser maior que zero para que o algoritmo funcione; então η_{ij} passou a ser uma função definida por partes na equação (16). Caso o valor de $g(ij)$ seja maior que zero esse mesmo número

representará a atratividade; caso seja igual a zero, então a visibilidade fica igual a 1; finalmente, caso esse valor seja negativo, arcos estão em atraso, então usa-se um valor entre 0 e 1. Esse cálculo acontece para todas as possibilidades que respeitam os movimentos proibidos pelos dois grafos esparsos criados e acontecem a cada decisão (para cada dia e carro). A informação heurística traz conhecimento específico da forma do problema e o cálculo em tempo real é importante para problemas dinâmicos (DORIGO; STÜTZLE, 2010).

A probabilidade também sofre interferência da experiência das rotas passadas usando os rastros de feromônio deixado por formigas durante as $|T|$ iterações. No início, em uma iteração $t = 0$, a quantidade de feromônios é baixa e a mesma para todos os arcos. Começando o algoritmo, cada formiga espalhará feromônios dos caminhos usados, então cada arco (i, j) atravessado receberá uma quantidade de feromônio. Conforme o tempo flui e as iterações passam o feromônio também sofre evaporação. $\tau_{ij}(t)$ é a informação numérica que representa a quantidade de feromônio em um arco (i, j) em uma iteração t . Quanto mais usado um arco, mais atrativo ele será para as próximas formigas. Existem diferentes maneiras de calcular a probabilidade de atravessar um arco, a mais usada é a regra de Sistemas de Formigas, *Ant System* (AS) (DORIGO; MANIEZZO; COLONI, 1996), a equação (17) representa a probabilidade.

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{j=1}^n [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} \quad \begin{array}{l} \forall [i, j] \in E, \\ t = 0, \dots, |T| \end{array} \quad (17)$$

A probabilidade de uma formiga sair de um nó i para um nó j em uma iteração t , $P_{ij}(t)$, resulta dos feromônios e da informação heurística (visibilidade). Ela mostra quanto realmente atrativo é um arco, α representa a importância dada para os feromônios e β para a informação heurística. Um sorteio aleatório decide o movimento de cada formiga seguindo a probabilidade da equação (17).

Com esse processo, a formiga move-se de camada para camada até formar a solução completa. Caso existam dois carros ou mais, o algoritmo gera um novo grafo para cada carro com um novo ponto inicial quando necessário. Se um carro inicia uma rota em um determinado ponto e termina ela no mesmo ponto, então o próximo carro deve selecionar um novo ponto para começar. Essa seleção é

aleatória e deve avaliar os arcos não bloqueados pelo grafo esparso e selecionar com uma maior probabilidade um nó que tenha mais arcos ligados que ainda não foram avaliados. Caso um carro termine sua rota em um ponto diferente do inicial, então o próximo carro deve começar os movimentos obrigatoriamente no ponto final do último carro para garantir a ciclicidade do problema.

4.3 A FASE DE MELHORIA

É comum hibridizar o ACO com outras heurísticas para melhorar a sua performance. Uma vez que uma solução completa é obtida, um procedimento de Busca Local, *Local Search* (LS), pode melhorá-la. As duas abordagens são complementares, o ACO explora o espaço solução de uma maneira “grosseira” enquanto a LS refina a solução encontrada. Essa combinação é fundamental para atingir resultados de estado-da-arte. Neste trabalho, foi desenvolvido um algoritmo de busca local simples que avalia duas vizinhanças e iterativamente melhora a solução com o melhor vizinho encontrado entre todos, regra da melhor melhoria.

A primeira vizinhança definida é a do já conhecido algoritmo 2-opt aplicado em diversos problemas de roteamento (TOTH; VIGO, 2003). Cada par de arcos não consecutivos é removido da solução e então um procedimento de reconstrução gera uma rota diferente: dois arcos novos ligam o trecho invertendo seu sentido. No PCARP-CM isso também significa mudar os dias de atendimento de um arco, de modo que essa operação é aplicada individualmente para cada carro. É possível aplicar um 2-opt entre rotas, porém no caso desse problema, isso implicaria uma dificuldade extra já que acarretaria na mudança do tamanho das rotas e violação da capacidade dos carros, então ele não foi utilizado.

A segunda vizinhança consiste na remoção de cada três arcos consecutivos e conseguinte avaliação de uma melhor maneira de reconstruir o trecho destruído. Essa remoção significa deixar de atender dois vértices (cidades) consecutivamente na solução, de modo que a reconstrução determina a substituição desses por um par diferente, novos arcos. Isso permite a eliminação de movimentos desnecessários e a inclusão de um arco não atendido respeitando sempre os pontos iniciais e finais de cada rota. Se o ponto final de algum carro é o inicial do próximo carro, então também é possível avaliar a troca desse por outro ponto.

4.4 O ALGORITMO COMPLETO

Sendo m o número de formigas, cada uma fará o movimento de um nó para outro com uma probabilidade dada por (17), respeitando os movimentos proibidos nos grafos esparsos quando possível. Assim que uma solução completa é obtida, ela é melhorada pelo algoritmo de busca local. Todos os arcos visitados receberão uma quantidade de feromônio deixada pelas formigas ao final de cada iteração. Na sequência, após todas as formigas gerarem suas respectivas soluções e despejarem feromônio, parte dele evapora seguindo uma taxa, ou seja, os feromônios em cada arco devem ser atualizados para as próximas iterações.

Cada solução completa e melhorada gerada por uma formiga gerará uma quantidade de feromônio a ser despejado entre as iterações t e $t + 1$. Vale lembrar aqui que o valor da função objetivo do problema em análise é composto por penalidade devido a atrasos e custos de deslocamento. Como o principal objetivo do problema é evitar com que atrasos ocorram, foi usado o número de atrasos em uma solução k , dado por nd^k , como critério para definir a quantidade de feromônio a ser despejada em cada trecho dada pela equação (18).

$$\Delta\tau_{ij}^k(t, t + 1) = \begin{cases} \frac{1}{nd^k} & \text{se } nd^k > 0 \text{ e a } k^a \text{ solução usa o arco } (i, j) \\ \frac{1}{\psi} & \text{se } nd^k = 0 \text{ e a } k^a \text{ solução usa o arco } (i, j) \\ 0 & \text{caso contrário} \end{cases} \quad (18)$$

$$\forall [i, j] \in E, k \in K$$

$$t = 0, \dots, |T| - 1$$

A equação (18) calcula a variação de feromônios entre as iterações causada pela k -ésima solução. Quanto mais atrasos, menos feromônios são deixados, uma solução desejável não deve conter penalidades e mais feromônios irão atrair mais formigas no futuro. Quando isso acontece usou-se um valor de $\psi = 0.1$, porém poderia ser usado qualquer valor no intervalo $[0, 1]$. A quantidade final de feromônio que restará para a próxima iteração, equação (19), resulta de todas as m soluções e um valor para $\rho \in [0, 1]$, onde $(1 - \rho)$ representa o coeficiente de evaporação.

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t, t+1) \quad \begin{matrix} \forall [i, j] \in E \\ t = 0, \dots, |T| - 1 \end{matrix} \quad (19)$$

O ALGORITMO 1 apresenta o procedimento completo. Primeiramente tem-se a entrada de dados com todas as informações relevantes ao problema. Todas as formigas irão usar o mesmo ponto inicial, qualquer um é um possível candidato, porém manter um ponto ajuda na convergência da solução e elimina soluções simétricas. Cada formiga fará sua rota selecionando movimentos para cada dia em cada carro. A solução gerada pela formiga será melhorada com o Algoritmo de Busca Local, caso um melhor resultado seja encontrado então a melhor solução encontrada é atualizada. A quantidade de feromônios a ser despejada é computada e depois que todas as formigas têm suas rotas prontas os feromônios são atualizados. O processo se repete até que algum critério de parada seja satisfeito sendo ele um tempo limite ou número de iterações.

ALGORITMO 1 - ACO PARA O PCARP-CM

1:	procedimento Recebe os dados ACO(carros, arcos, cidades)
2:	Selecionar um ponto inicial
3:	Faça até alcançar um critério de parada
4:	Para cada formiga // ($i = 1$ até m)
5:	Para cada carro // ($k = 1$ até K)
6:	Se carro $\neq 1$ então escolha o ponto inicial
7:	Carregue os possíveis pontos finais no último dia para o carro k
8:	Gere o grafo para o carro
9:	Para cada dia // ($d = 1$ até np)
10:	Calcule a visibilidade η_{ij} para cada movimento possível
11:	Sorteie a próxima cidade a ser visitada
12:	Atualize o grafo esparso
13:	Aplique o procedimento de melhoria (Busca Local)
14:	Se a solução é a melhor então salve
15:	Calcule a quantidade de feromônio a ser despejado pela formiga i em cada arco
16:	Atualize os feromônios utilizando as variações e o coeficiente de evaporação ρ
17:	Fim faça
18:	Retorne a melhor solução encontrada

FONTE: O autor (2018).

4.5 DEFINIÇÃO DE PARÂMETROS PARA O ALGORITMO ACO

O ACO utiliza alguns parâmetros que dizem respeito a relativa importância dada para o rastro de feromônios $\alpha \geq 0$, para a visibilidade ou informação heurística $\beta \geq 0$ combinado com um coeficiente $0 \leq \rho \leq 1$ que determina a evaporação. Para determiná-los rodou-se 29250 testes combinando alguns valores promissores usados em Colorni, Dorigo e Maniezzo (1991) e em Dorigo, Maniezzo e Colorni (1996) que foram: $\alpha = \{0,5; 1; 1,2; 2; 5\}$, $\beta = \{0,5; 1; 2; 3; 5\}$ e $\rho = \{0,5; 0,7; 0,9\}$. Um número significativo de formigas permite a exploração de mais soluções e gera mais informações para atualizar os feromônios, por este motivo foi utilizada uma quantidade de formigas $m = 100$.

TABELA 4.1 - RESULTADOS PARA DEFINIÇÃO DOS PARÂMETROS DO ACO

$\alpha; \beta; \rho$	Média	Desv. Pad.	$\alpha; \beta; \rho$	Média	Desv. Pad.	$\alpha; \beta; \rho$	Média	Desv. Pad.
0,5;0,5;0,5	15859,2	13475,7	1;3;0,7	14156,6	12647,3	2;1;0,9	16600,6	13848,6
0,5;0,5;0,7	16183,0	13854,6	1;3;0,9	15025,6	13005,7	2;2;0,5	18858,3	15455,2
0,5;0,5;0,9	16865,6	14362,9	1;5;0,5	13867,7	12361,2	2;2;0,7	18088,9	14983,7
0,5;1;0,5	16338,2	13864,6	1;5;0,7	14220,6	12677,9	2;2;0,9	16895,0	14258,3
0,5;1;0,7	16366,0	13904,9	1;5;0,9	15311,4	13212,6	2;3;0,5	18889,3	15366,4
0,5;1;0,9	16448,7	13743,4	1,2;0,5;0,5	14694,5	12995,8	2;3;0,7	18308,4	15481,3
0,5;2;0,5	16179,6	13586,8	1,2;0,5;0,7	14017,7	12392,5	2;3;0,9	16936,0	13945,6
0,5;2;0,7	16568,2	13880,8	1,2;0,5;0,9	13964,2	12529,1	2;5;0,5	19562,6	16174,2
0,5;2;0,9	16399,8	13921,1	1,2;1;0,5	14887,4	13169,0	2;5;0,7	18496,9	14940,5
0,5;3;0,5	16186,0	13569,8	1,2;1;0,7	13984,9	12547,5	2;5;0,9	16932,2	13884,1
0,5;3;0,7	16282,8	13841,7	1,2;1;0,9	14585,2	13018,0	5;0,5;0,5	22404,5	18139,3
0,5;3;0,9	16575,3	14138,1	1,2;2;0,5	14826,1	12995,7	5;0,5;0,7	21988,9	17097,7
0,5;5;0,5	16400,4	13687,8	1,2;2;0,7	14220,1	12561,7	5;0,5;0,9	21437,5	16956,4
0,5;5;0,7	16543,2	14035,1	1,2;2;0,9	14482,6	12947,9	5;1;0,5	22394,5	18020,2
0,5;5;0,9	16782,8	13877,7	1,2;3;0,5	14689,6	13037,8	5;1;0,7	22074,8	17453,2
1;0,5;0,5	13692,0	12598,6	1,2;3;0,7	14149,5	12313,9	5;1;0,9	21361,2	16944,4
1;0,5;0,7	14824,3	13243,7	1,2;3;0,9	14387,4	12872,5	5;2;0,5	22379,4	17334,7
1;0,5;0,9	14973,1	13057,2	1,2;5;0,5	14964,4	13208,6	5;2;0,7	21950,8	16971,1
1;1;0,5	13218,2	12314,9	1,2;5;0,7	14426,6	12720,8	5;2;0,9	21665,8	17300,3
1;1;0,7	14019,0	12449,8	1,2;5;0,9	14424,1	12987,1	5;3;0,5	22677,1	17584,1
1;1;0,9	15032,9	13283,5	2;0,5;0,5	19080,2	15207,4	5;3;0,7	21998,5	17490,6
1;2;0,5	13334,1	11985,9	2;0,5;0,7	18460,9	15324,0	5;3;0,9	21537,6	16456,3
1;2;0,7	14036,5	12858,2	2;0,5;0,9	16795,2	13861,4	5;5;0,5	22644,0	17977,5
1;2;0,9	15299,7	13331,7	2;1;0,5	19165,3	15803,9	5;5;0,7	21955,4	16996,4
1;3;0,5	13567,2	11982,5	2;1;0,7	18345,1	15311,8	5;5;0,9	21402,6	16361,2

FONTE: o autor (2018)

Para determinar os pesos de α , β e ρ se usou um critério de parada de 100 iterações. Nesse teste, a busca local foi eliminada do algoritmo, linha 13 do ALGORITMO 1, pois é um procedimento que pode tomar um tempo considerável quando a solução usada é pobre. Com essa remoção também pode-se encontrar parâmetros que têm uma melhor convergência sem precisar da melhoria o que poderia tornar o processo mais eficiente. Cada instância foi testada 10 vezes, resultando em 390 testes para cada combinação de parâmetros. Cada teste levou em média 32 segundos para ser executado completamente e os resultados estão na TABELA 4.1.

A quantidade de 10 testes para cada combinação de parâmetros foi feita no intuito de avaliar possíveis oscilações do método. Os parâmetros que apresentam o menor valor em média para todos são interessantes para a aplicação do algoritmo, pois atingem de maneira geral os melhores resultados, além disso o desvio padrão também se mostrou baixo em comparação aos demais. A TABELA 4.1 mostra que a melhor combinação de parâmetros a ser usada é composta por $\alpha = 1$, $\beta = 1$ e $\rho = 0.5$. Também foi analisada cada instância individualmente e cada uma tem um parâmetro que se destaca em performance do algoritmo, porém não é visível uma relação entre características do problema e bons parâmetros.

4.6 RESULTADOS ALCANÇADOS

Alguns testes foram feitos com o algoritmo ACO para ver sua performance, principais características e pontos fortes. Inicialmente estabeleceu-se como critério de parada um tempo limite de 3600 segundos (1 hora), de maneira que quando esse tempo fosse atingido o algoritmo pudesse terminar a iteração que está presente. Como já mencionado, o algoritmo de Busca Local pode tomar um tempo longo e o critério de parada estabelecido faz com que ao menos uma iteração completa seja feita. Para critérios de comparação, levantou-se o resultado obtido em cada instância no *so/ver* Gurobi 8.0.1 após uma hora de resolução nos testes da seção 3.4. Os resultados estão presentes na TABELA 4.2.

TABELA 4.2 - COMPARAÇÃO SOLVER VERSUS ACO EM 1 HORA

Instância	Solver em 1 hora	ACO			Taxa de Ocup. Mínima	Melhor
		Valor da FO	Iterações	Tempo (s)		
<i>pgdbcm1</i>	19416	451	312	3606	84,21%	ACO
<i>pgdbcm2</i>	23564	3594	10	3680	83,02%	ACO
<i>pgdbcm3</i>	340	344	89	3628	81,58%	Solver
<i>pgdbcm4</i>	9538	2519	44	3644	80,95%	ACO
<i>pgdbcm5</i>	17773	9764	12	3832	82,81%	ACO
<i>pgdbcm6</i>	456	456	312	3606	80,95%	Empate
<i>pgdbcm7</i>	549	576	32	3675	27,41%	Solver
<i>pgdbcm8</i>	20395	22364	4	3870	82,72%	Solver
<i>pgdbcm9</i>	54419	31446	3	4492	81,25%	ACO
<i>pgdbcm10</i>	1741	3801	4	3839	83,33%	Solver
<i>pgdbcm11</i>	30465	442	10	3737	83,33%	ACO
<i>pgdbcm12</i>	545	536	306	3606	82,50%	ACO
<i>pgdbcm13</i>	620*	620	18	3704	82,50%	Empate
<i>pgdbcm14</i>	145*	149	12	3742	80,00%	Solver
<i>pgdbcm15</i>	144	1152	4	4010	82,76%	Solver
<i>pgdbcm16</i>	182	204	2	5796	55,56%	Solver
<i>pgdbcm17</i>	175	183	3	4963	82,54%	Solver
<i>pgdbcm18</i>	9410	2447	1	10322	81,82%	ACO
<i>pgdbcm19</i>	125*	131	45	3632	60,42%	Solver
<i>pgdbcm20</i>	14265	4278	8	3939	82,81%	ACO
<i>pgdbcm21</i>	212	222	5	4013	76,67%	Solver
<i>pgdbcm22</i>	482	1492	1	13001	80,00%	Solver
<i>pgdbcm23</i>	340	357	1	18064	80,39%	Solver
<i>pgdbcm24</i>	8586	620	10	3927	81,48%	ACO
<i>pgdbcm25</i>	348	370	127	3612	77,50%	Solver
<i>pgdbcm26</i>	548	573	39	3602	70,83%	Solver
<i>pgdbcm27</i>	723	4789	11	3898	78,13%	Solver
<i>pgdbcm28</i>	444	447	30	3674	68,75%	Solver
<i>pgdbcm29</i>	2405	3415	3	4581	70,00%	Solver
<i>pgdbcm30</i>	46441	25424	3	4625	81,25%	ACO
<i>pgdbcm31</i>	4838	4906	3	4621	72,92%	Solver
<i>pgdbcm32</i>	527	536	208	3600	78,57%	Solver
<i>pgdbcm33</i>	157	158	13	3638	90,00%	Solver
<i>pgdbcm34</i>	120	131	4	4512	70,00%	Solver
<i>pgdbcm35</i>	174	209	2	5145	75,00%	Solver
<i>pgdbcm36</i>	383	2479	1	13780	75,00%	Solver
<i>pgdbcm37</i>	1284	4284	8	3969	36,81%	Solver
<i>pgdbcm38</i>	427	525	1	16899	73,33%	Solver
<i>pgdbcm39</i>	338	380	1	22137	75,93%	Solver
Média	7001	3507				

*Solução ótima

FONTE: o autor (2018)

A primeira coluna da TABELA 4.2 mostra qual a instância abordada. A segunda, o valor alcançado pelo *sol/ver* após uma hora. As três colunas seguintes trazem os resultados do algoritmo ACO: a primeira dessas colunas mostra o valor da função objetivo, a segunda o número de iterações e a terceira o tempo computacional. Vale ressaltar que o algoritmo teve alto tempo computacional em algumas instâncias em virtude da busca local e a maneira com que os critérios de parada foram estabelecidos. A sexta coluna mostra uma taxa de ocupação mínima dos veículos que pode ser calculada pela equação (20). Já a última coluna mostra qual abordagem de resolução obteve melhor resultado. A última linha mostra o valor médio alcançado em todas instâncias, sendo possível observar que o resultado médio do Algoritmo ACO chegou a ser quase a metade do valor obtido pelo *sol/ver* comercial, mostrando a competitividade do algoritmo proposto.

$$\text{Taxa de ocupação mínima} = \frac{\text{número de necessidades}}{\text{número de carros} \times \text{horizonte de tempo}} \quad (20)$$

Em contraponto ao valor médio apresentado, a TABELA 4.2 mostra a superioridade do *sol/ver* para muitas das instâncias. Sabendo que as soluções com valores de 3 casas decimais são respostas em que não há atrasos nas periodicidades, pode-se concluir que quando o valor das funções objetivo é dessa dimensão a superioridade do *sol/ver* contra o ACO é pequena. Agora, quando o valor das funções objetivo é maior, o que significa que há punições, o ACO consegue superar com maior diferença os valores do *sol/ver*. A equação (20) que representa a taxa de ocupação mínima dos veículos mostra a liberdade dos carros para fazer movimentos de deslocamento sem a necessidade de fazer medições. Se avaliadas situações críticas nas quais os carros devem ser ocupados em mais de 80% de sua capacidade para executar suas tarefas, o ACO consegue atingir melhores resultados em 13 de 20 instâncias, ou seja, em 65% das vezes. Isso mostra um ponto forte do algoritmo, que consegue trabalhar muito bem nessas situações críticas. Já para situações onde existe mais liberdade para os carros, o *sol/ver* foi imbatível. Esse resultado pode ser observado na TABELA 4.3.

Uma prática possível com os *sol/vers* é tentar fazê-los alcançar melhores resultados dando a eles uma solução inicial para iniciar o processo. Esse procedimento não dá garantia de que a convergência de um problema seja mais

rápida, porém pode ajudar bastante. Em algumas situações, o *solver* pode demorar para encontrar uma primeira solução factível e, tendo uma pronta, ele pode melhorá-la; caso seja a ótima, talvez fazer a prova.

TABELA 4.3 - PERFORMANCE *SOLVER* E ACO SEGUNDO TAXA DE OCUPAÇÃO MÍNIMA

Taxa de Ocupação Mínima	Número de instâncias	Melhores performances		
		<i>Solver</i>	ACO	Empate
> 80%	20	7	11	2
≤ 80%	19	19	0	0

FONTE: o autor (2018)

O teste seguinte visa alcançar resultados melhores para os 39 problemas teste, instâncias, com um limitante de tempo de resolução de 24 horas, ou seja, 86400 segundos usando a resposta obtida com a ACO como solução inicial. Como o tempo dentro de uma operação de busca local pode ser muito longo, limitou-se o algoritmo ACO a trazer a resposta disponível após 1 hora, independentemente de uma iteração ter sido completa ou não. Logo, o tempo disponível para otimização no Gurobi é de 23 horas. A avaliação do tempo como critério de parada foi estabelecida dentro da operação de busca local, linha 13 do ALGORITMO 1. Os resultados obtidos com esse teste estão à disposição na TABELA 4.4.

A primeira coluna apresenta as instâncias e as próximas 3 colunas os dados dos testes apresentados na seção 3.4. A seguinte mostra o valor obtido com o ACO, enquanto as 4 seguintes mostram os resultados alcançados pelo *solver* com o ACO passando a solução inicial. Finalmente, as últimas duas colunas trazem o melhor resultado até o momento. Como a modelagem é a mesma, o melhor *lower bound* encontrado é o maior valor entre os dois *lower bounds* com destaque em negrito. Sendo o problema de minimização, o melhor valor para uma solução é o menor valor, também em negrito.

É possível notar que a melhor solução não foi conseguida nessa estratégia proposta em apenas três instâncias: 4, 26 (que teve origem em 4) e a 27. A estratégia ainda encontrou 13 resultados melhores dos que haviam sido apresentados nas instâncias: 2, 5, 8, 9, 10, 18, 20, 22, 24, 30, 31, 36 e 37. É possível ver também que o *lower bound* aumentou em 26 instâncias, significando que o valor mínimo esperado é maior do que o que havia até então, ficando mais próximo de provar otimalidade. Em contrapartida, a estratégia com solução inicial não conseguiu provar dois ótimos que já eram conhecidos nas instâncias 32 e 34.

TABELA 4.4 - RESULTADOS DO SOLVER UTILIZANDO SOLUÇÃO INICIAL DADA PELO ALGORITMO ACO

Instâncias	Solver				Solver com solução inicial				Melhores		
	Lower bound	Solução	Tempo (seg.)	Valor do ACO em 1 hora	Lower bound	Solução	GAP	Tempo (seg.)	Lower Bound	Solução	Gap
<i>pgdbcm1</i>	451	451	29869	1451	451	451	0%	16937	451	451*	0%
<i>pgdbcm2</i>	518	614	86400	605	519	582	11%	82800	519	582	11%
<i>pgdbcm3</i>	315	340	86401	340	331	340	3%	82800	331	340	3%
<i>pgdbcm4</i>	449	2519	86401	3527	453	2546	82%	82800	453	2519	82%
<i>pgdbcm5</i>	659	6737	86400	10760	659	5742	89%	82800	659	5742	89%
<i>pgdbcm6</i>	418	456	86401	456	418	456	8%	82801	418	456	8%
<i>pgdbcm7</i>	494	508	86400	583	496	508	2%	82801	496	508	2%
<i>pgdbcm8</i>	325	8395	86400	19368	325	5385	94%	82802	325	5385	94%
<i>pgdbcm9</i>	439	44425	86401	34437	439	30443	99%	82803	439	30443	99%
<i>pgdbcm10</i>	684	766	86400	5779	684	733	7%	82802	684	733	7%
<i>pgdbcm11</i>	416	442	86400	489	413	442	7%	82804	416	442	6%
<i>pgdbcm12</i>	484	536	86400	536	481	536	10%	82803	484	536	10%
<i>pgdbcm13</i>	620	620	10	620	620	620	0%	40	620	620*	0%
<i>pgdbcm14</i>	145	145	33	151	145	145	0%	4	145	145*	0%
<i>pgdbcm15</i>	137	140	86400	152	137	140	2%	82800	137	140	2%
<i>pgdbcm16</i>	176	182	86402	200	176	182	3%	82802	176	182	3%
<i>pgdbcm17</i>	161	168	86400	2184	161	168	4%	82800	161	168	4%
<i>pgdbcm18</i>	369	379	86400	2440	369	377	2%	82801	369	377	2%
<i>pgdbcm19</i>	125	125	109	131	125	125	0%	1291	125	125*	0%
<i>pgdbcm20</i>	240	2262	86400	5291	247	258	4%	82800	247	258	4%
<i>pgdbcm21</i>	210	212	86402	221	206	212	3%	82803	210	212	1%
<i>pgdbcm22</i>	423	435	86400	2485	423	426	1%	82800	423	426	1%
<i>pgdbcm23</i>	329	336	86402	369	329	336	2%	82803	329	336	2%
<i>pgdbcm24</i>	517	582	86400	1669	520	576	10%	82800	520	576	10%
<i>pgdbcm25</i>	323	340	86400	340	323	340	5%	82801	323	340	5%
<i>pgdbcm26</i>	447	508	86400	552	448	525	15%	82801	448	508	12%
<i>pgdbcm27</i>	639	676	86400	3830	639	680	6%	82800	639	676	5%
<i>pgdbcm28</i>	402	444	86405	449	418	444	6%	82802	418	444	6%
<i>pgdbcm29</i>	305	339	86400	3418	305	339	10%	82800	305	339	10%
<i>pgdbcm30</i>	375	21449	86401	21419	375	8442	96%	82800	375	8442	96%
<i>pgdbcm31</i>	684	748	86400	3888	684	737	7%	82800	684	737	7%
<i>pgdbcm32</i>	523	523	73300	545	491	523	6%	82800	523	523*	0%
<i>pgdbcm33</i>	153	157	86400	1163	153	157	3%	82800	153	157	3%
<i>pgdbcm34</i>	120	120	57310	136	117	120	2%	82800	120	120*	0%
<i>pgdbcm35</i>	167	170	86400	207	167	170	2%	82801	167	170	2%
<i>pgdbcm36</i>	369	381	86400	2468	369	371	1%	82800	369	371	1%
<i>pgdbcm37</i>	240	264	86400	5298	244	261	7%	82800	244	261	7%
<i>pgdbcm38</i>	423	426	86400	541	423	426	1%	82800	423	426	1%
<i>pgdbcm39</i>	329	336	86403	374	329	336	2%	82802	329	336	2%

*Solução ótima

FONTE: o autor (2018)

4.7 CONSIDERAÇÕES SOBRE O ALGORITMO ACO APLICADO AO PCARP-CM

O algoritmo da colônia de formigas encontra instantaneamente uma solução factível qualquer a partir do momento que uma formiga completa seus deslocamentos. Como observado na seção 4.5, 100 iterações sem a utilização da busca local demoram em média 32 segundos, ou seja, a cada 32 segundos são criadas e avaliadas 10000 rotas. Em contrapartida, o processo de melhoria pode demorar, visto que as soluções são muito diversificadas e que quando a entrada para a busca local é uma solução pobre, o algoritmo gasta mais tempo melhorando-a. Sem o algoritmo de melhoria, o desempenho do ACO é muito fraco.

A aplicação do ACO mostra seu ponto forte em problemas críticos que tendem a ter atrasos. A hibridização usando ACO como um criador de soluções iniciais para o *solver* mostrou-se uma boa estratégia, alcançando os melhores desempenhos em 36 das 39 instâncias, ou seja, em 92,3% das vezes. O algoritmo ACO é muito rico e traz uma série de informações como, por exemplo, quais são os arcos mais atrativos em dados períodos. Essa informação é valiosa e pode ser usada como fonte para outros tipos de hibridização com modelos exatos. Uma técnica capaz de fazer isso é apresentada no próximo capítulo.

5 ITERATIVE RESTRICTED SPACE SEARCH APLICADO AO PROBLEMA DE ROTEAMENTO PERIÓDICO E CAPACITADO EM ARCOS COM MOVIMENTOS CONTÍNUOS

Essa seção visa detalhar o funcionamento do método de Busca Iterativa em Espaços Restritos, *Iterative Restricted Space Search* (IRSS), proposto em Pécora Jr. (2008), todo seu detalhamento pode ser encontrado na referenciada tese. O algoritmo foi inspirado no algoritmo *Local Branching* (FISCHETTI; LODI, 2003) ao fazer uma busca em uma árvore binária controlando subespaços do espaço solução a serem explorados iterativamente. Além disso, esse capítulo tem como objetivo mostrar qual foi a postura tomada para adaptar o algoritmo ao PCARP-CM usando hibridização com o ACO. Por fim, são apresentados resultados alcançados pelo IRSS.

5.1 ITERATIVE RESTRICTED SPACE SEARCH – IRSS

A busca por uma solução ótima em um espaço solução de um problema grande pode parecer muitas vezes uma tarefa intratável. Quanto mais complexos os problemas, melhores devem ser as técnicas a ele aplicadas de forma a garantir uma eficiente resolução. Uma alternativa é a hibridização de técnicas de resolução a fim de somar forças para resolver problemas, os pontos fortes de cada técnica são listados e mesclados formando um novo algoritmo. O IRSS é um algoritmo que se baseia em uma troca de informações entre heurísticas e métodos exatos gerando um “*framework*” possível de aplicar em diversos problemas.

“A ideia principal do IRSS é iterativamente definir e explorar regiões restritas – subconjuntos do espaço solução global – que tenham alto potencial de conter boas soluções e são pequenas o suficiente para serem completamente exploradas. Essas regiões recebem o nome de espaços restritos. O IRSS é uma abordagem genérica de busca baseada na interação de duas fases algorítmicas tendo objetivos complementares. A primeira fase identifica o espaço restrito e a segunda o explora. O algoritmo alterna entre as duas fases por um número fixo de iterações ou até que um tempo alocado seja expirado.” (PÉCORA JR, 2008, p. iv, tradução nossa)

A tese de Pécora Jr. (2008) é dividida em 4 partes. A primeira parte explora a utilização de métodos híbridos e suas aplicações; a segunda apresenta a aplicação de métodos híbrido-exatos em um problema de alocação de recurso em

usinas de celulose; a terceira mostra como é possível evitar a exploração de regiões já avaliadas em um problema de localização e alocação; e a quarta solidifica de forma generalizada os conceitos do *framework*, passível de ser aplicada em diversos problemas.

Algoritmos de busca se mostram muito eficientes trabalhando em espaços solução pequenos. Logo, limitar o espaço solução de um problema, reduzindo-o ou restringindo-o pode ser um caminho para encontrar boas soluções e talvez até a ótima. A criação desse método foi baseada no paradigma de diversificação e intensificação dos algoritmos, diversificar é conseguir alternar entre espaços e intensificar é buscar dentro do espaço. Um espaço restrito segue a definição:

“Espaço Restrito, *Restricted Space* (RS), é um subespaço do conjunto universo de soluções que contém duas características altamente desejáveis: (1) deve ser pequena o suficiente para ser completamente explorada e (2) deve ter uma alta possibilidade de conter soluções próximas da ótima.” (PÉCORA JR, 2008, p. 2, tradução nossa)

O Espaço Restrito, *Restricted Space* (RS), é uma região que contém um subconjunto de soluções com ótimos locais, que podem vir a ser ótimos globais do problema. A identificação das regiões e a limitação de cada uma delas é uma das tarefas mais desafiadoras do algoritmo. Ela depende de algum algoritmo auxiliando a limitação e que de preferência contenha as seguintes características:

1. Explore, o quanto possível, o espaço solução;
2. Tenha informações o suficiente para identificar características comuns às boas soluções, não necessariamente precisando encontrar a melhor delas;
3. Faça um ranqueamento de possíveis regiões por meio de uma função objetivo;
4. Eficiência.

O IRSS é um algoritmo híbrido, ou seja, uma mistura de ao menos duas técnicas que resolvem um problema. Muitos artigos apresentam hibridização entre heurísticas, porém a hibridização com métodos exatos está em alta visto a capacidade dos *solvers* atuais. A utilização de métodos exatos é adequada na resolução de problemas pequenos ou médios, porém muitas vezes é custosa para problemas de grande porte e que sejam *NP-Hard*. Os métodos exatos podem garantir a otimalidade de um problema, enquanto heurísticas ou metaheurísticas não

conseguem fazer essa prova. O ponto forte das heurísticas e metaheurísticas estão relacionados à eficiência em alcançar bons resultados, logo formam um bom leque de opções para serem combinados com o IRSS.

O trabalho restringindo regiões do espaço solução é recorrente em Pesquisa Operacional e acaba sendo uma alternativa na resolução de problemas grandes. Exemplos podem ser encontrados quando algum problema grande é dividido formando subproblemas de pequena escala, estratégia de dividir para conquistar. Outros tipos de estratégia são geração de colunas, *cutting plane algorithms*, relaxação linear, etc. A conversão de problema principal em outro subproblema menor ou mais administrável/resolvível pode ser feita com manipulações ou aproximações matemáticas (GEOFFRION, 1970a, 1970b).

É importante ressaltar que essas manipulações matemáticas acabam afetando o espaço de soluções e podem não permitir que a solução ótima seja alcançada. A utilização de heurísticas baseadas em vizinhança como Busca Tabu, *Simulated Annealing* e *Variable Neighborhood Search* também trabalham em regiões restritas, porém não modificam o espaço de soluções. A utilização de parâmetros de diversificação como a Lista Tabu é algo necessário nos algoritmos para conseguir atingir vizinhanças diferentes.

A ideia de fazer a busca em um espaço restrito é aplicada em um problema chamado *Pulp Production Scheduling Problem* (PPSP), Problema de Produção e sequenciamento de Celulose (PÉCORA JR; RUIZ; SORIANO, 2007). A celulose, matéria prima da indústria do papel, é obtida mediante um processo de cozimento da madeira. Um bom processo faz o cozimento das madeiras de mesma densidade juntas, logo um planejamento de onde colher e onde alocar madeiras para o processo, otimiza o serviço. Nesse problema, o espaço de busca foi limitado gerando apenas um espaço restrito.

O algoritmo IRSS propriamente dito, o qual usa uma árvore binária que controla a busca dentro de cada espaço restrito, é aplicado a um problema de Localização e Alocação no contexto de uma companhia de transporte internacional. A utilização de uma árvore binária permite que a busca seja feita em mais de uma Região Restrita, logo ela é responsável pela diversificação do espaço de busca. Além disso, o controle permite que a busca seja feita em regiões novas, de forma que a interseção de duas regiões restritas resulta em um conjunto vazio.

O problema de localização e alocação tratado em Pécora Jr. (2008) é composto por custos não lineares de aberturas de novos *hubs* (espécie de depósito) e custos de transporte até novos clientes, sendo possível a economia de escala. A necessidade de abrir um *hub*, dimensionar seu tamanho e definir o caminho de cada transporte fazem com que a abordagem por meio de um método exato seja ineficiente.

O algoritmo IRSS trabalha com duas fases. A primeira fase é a de restrição e consiste na definição de um Espaço Restrito que deve ser completamente explorado na segunda fase, logo, a segunda fase é uma busca dentro desse espaço limitado. O IRSS alterna entre essas duas fases a fim de avaliar uma quantidade razoável de espaços restritos distintos. Quando a solução ótima é encontrada durante a fase de busca, o algoritmo volta à primeira fase em busca de outro Espaço Restrito diferente dos anteriores. A segunda fase pode retornar para a primeira fase as melhores soluções encontradas, com o intuito de induzir a geração do próximo espaço restrito em uma região complementar a que já foi explorada. O algoritmo é finalizado após explorar um certo número de Espaços Restritos ou até expirar um tempo de processamento computacional estabelecido.

5.1.1 Detalhamento das fases

Como mencionado, as duas fases do algoritmo podem ser chamadas de fase de restrição e fase de busca. O detalhamento dessas fases será explicado nessa seção.

5.1.1.1 Fase de restrição

Para a fase de restrição pode não ser muito adequado a utilização de algoritmos baseados unicamente em busca local (PÉCORA JR., 2008). O ideal é usar algum algoritmo que explore uma grande região do espaço de soluções e que tenha uma visão macro do problema. As operações matemáticas que irão restringir um espaço usarão soluções ou informações obtidas por esse algoritmo.

A restrição pode ser feita usando algumas variáveis importantes para o problema, por exemplo, as variáveis que determinam as localizações a serem abertas/utilizadas no trabalho de Pécora Jr. (2008). Se for pré-determinado que

algumas variáveis assumirão obrigatoriamente um valor enquanto as demais variáveis ficam livres, o espaço de busca passa a ser reduzido. Dependendo da quantidade de variáveis usadas, a exploração ainda pode acontecer em uma grande porção do espaço solução.

As variáveis que devem ser estudadas nessa etapa são chamadas de estruturais, sendo algumas das variáveis mais importantes do modelo e que dão uma direção significativa para a resolução. As variáveis estruturais acabam limitando uma série de opções na resolução do problema, ou seja, algumas das demais variáveis também podem assumir valores limitados. O conceito de variáveis estruturais não pode ser generalizado e uma análise envolvendo a hierarquia e o nível das decisões combinado com a abordagem de solução usada devem indicar quais são essas variáveis (SCHNEEWEISS, 2003).

Os Espaços Restritos são o resultado da fase de restrição e podem ser vistos como envoltórios convexos, contendo soluções e que serão gerados de maneira ordenada. Uma das ideias para delimitar cada espaço é usar um *poll* de soluções geradas de maneira rápida e eficaz, nesse *poll* é possível identificar alguns padrões de resposta que se repetem para gerar a restrição. Um padrão se repetindo muitas vezes pode ser um indicativo de que ali está um pedaço de solução promissora a alcançar bons resultados.

A exemplificação por Pécora Jr. (2008) de um espaço restrito, quando trata do problema de localização e alocação, é da seguinte forma: dado um problema que trabalha com uma variável binária $z_{h,l}$ e supondo que por meio de uma heurística seja possível obter n soluções para o problema, tem-se que cada solução $s^i \in S$. Ou seja, S é o conjunto de soluções. Seja J o conjunto de todas as variáveis que assumem o valor de 0 em todas as n soluções e K o conjunto de todas as variáveis que assumem valor 1, formalmente definido como $J = \{(h, l) \mid \sum_i (1 - z_{h,l}^i) \geq |S|\}$ e $K = \{(h, l) \mid \sum_i z_{h,l}^i \geq |S|\}$. Por exemplo, supondo que $z_{1,2} = 1$ em todas as n soluções, então ele pertence a K . A restrição que gera a RS pode ser expressa como (21):

$$\sum_{j \in J} z_j + \sum_{k \in K} (1 - z_k) = 0 \quad (21)$$

“O tamanho da RS pode ser definido como sendo o número das variáveis não fixadas dentro da RS, $|RS| = |H| \times |L| - |J| - |K|$. Então, a RS pode ser vista como uma união de muitas soluções. Infelizmente, a definição dos conjuntos J e K pode levar a um espaço restrito muito grande que ainda pode não ser explorado completamente em um tempo razoável. Além disso, uma escolha desafortunada do conjunto de variável J e K pode acarretar em soluções desinteressantes. Para superar essas desvantagens, a abordagem IRSS pode ser aperfeiçoada com (1) uma abordagem de decomposição chamada de Espaços Restritos *Embedded* ou “Embutidos” que permite uma exploração “progressiva” do RS e (2) uma estrutura de árvore binária que permite a geração e vista de muitas regiões restritas distintas.” (PÉCORA JR, 2008, p. 93, tradução nossa)

A ideia de Espaços Restritos Embutidos procura particionar a RS, subdividindo-a em subespaços autocontidos com tamanhos cada vez menores. Isso permite a formação dos conjuntos J e K com variáveis que não necessariamente aparecem em 100% das vezes com valores iguais nas n soluções, porém em porcentagens menores variando de 0% a 100%. Quanto menor a porcentagem, maiores os conjuntos J e K serão e mais variáveis são fixadas, logo menores serão as regiões restritas para busca. Essa porcentagem é o que define um RS embutido, de maneira geral as alterações na porcentagem permitem a definição de uma RS de quase qualquer tamanho.

É importante ressaltar que quando um espaço restrito é infactível, todos os subespaços menores e contidos nele também são infactíveis. Quando um subespaço explorado chega em um resultado ótimo, todos os subespaços contidos nele terão funções objetivos iguais ou piores. Uma forma de administrar a porcentagem que será estabelecida formando um RS embutido é usando os valores das funções objetivo determinadas na fase anterior.

Embora a flexibilidade de usar espaços restritos embutidos de qualquer tamanho seja interessante e eficiente, é muito difícil determinar a porcentagem a ser usada. Uma melhoria importante proposta para IRSS (PÉCORA JR, 2008) é a utilização de Espaços Restritos Embutidos Dinâmicos: depois de cada iteração o tamanho do espaço embutido pode variar dependendo do tempo computacional de resolução de cada modelo.

5.1.1.2 Processo de busca e a estrutura de árvore binária

O processo desenvolvido por Pécora Jr. (2008) propõe uma partição binária das regiões de busca inspirada no artigo de Fischetti e Lodi (2003), que trata do método *Local Branching*. Sempre que é gerado um espaço restrito, é possível encontrar o complementar desse espaço. A expressão matemática do espaço complementar é dada pela inequação (22):

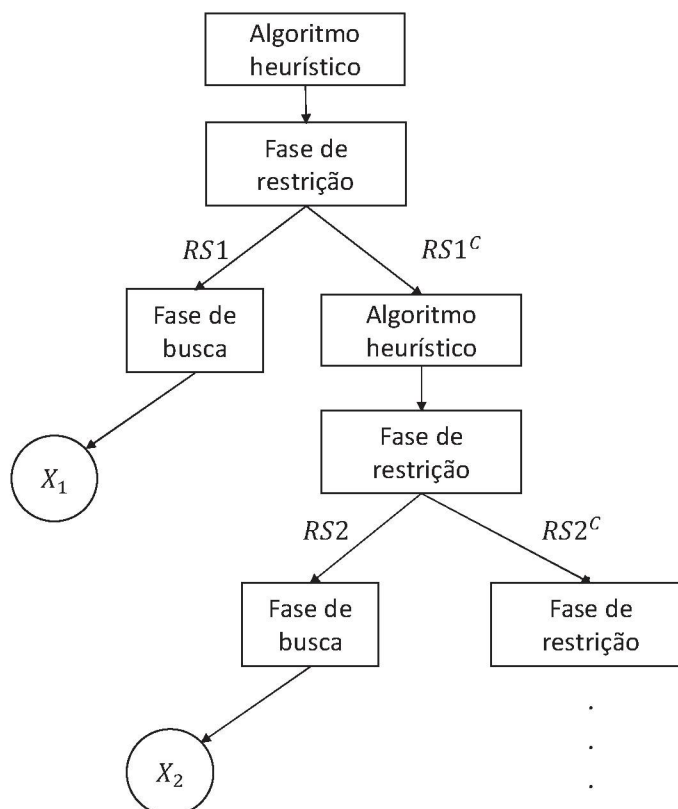
$$\sum_{j \in J} z_j + \sum_{k \in K} (1 - z_k) \geq 1 \quad (22)$$

A FIGURA 5.1 ilustra a utilização da árvore binária na resolução de um problema qualquer no qual existe uma interação entre as fases de restrição e busca. Esse esquema oferece a vantagem de explorar as soluções de forma intensificada e diversificada. Não é possível falar com precisão em qual RS estará a solução com valor ótimo global, porém com toda certeza ele se encontra em algum deles e a exploração na árvore pode permitir que ele seja encontrado. O algoritmo armazena na memória as regiões já avaliadas, evitando assim o desperdício de tempo e fazendo buscas em locais repetitivos, maximizando a eficiência do algoritmo.

A FIGURA 5.1 também mostra como funciona o processo, no qual um algoritmo heurístico gerará soluções e/ou informações que serão passadas para a fase de restrição. Na fase de restrição, iteração 1, o algoritmo vai dividir todo o espaço solução em duas regiões, ou seja, em Espaço Restrito 1 ($RS1$) e Espaço Restrito Complementar 1 ($RS1^C$). A $RS1$ no primeiro ramo esquerdo é completamente explorada pela fase de busca, trazendo o melhor resultado X_1 . Então o algoritmo deve ser aplicado novamente para auxiliar a nova fase de restrição que deve ser aplicada em $RS1^C$, identificando um Espaço Restrito 2 ($RS2$), vale notar que $RS2 \subset RS1^C$ e $RS1^C$ é dividida em $RS2$ e $RS2^C$. A segunda ramificação à esquerda ($RS2$) é completamente explorada trazendo uma solução ótima X_2 . O processo continua com novas fases de restrição dentro dos espaços inexplorados e novas fases de busca até que algum critério de parada seja satisfeito. As ramificações à esquerda são sempre resolvidas e geram uma solução enquanto as que estão à direita funcionam como memória para evitar a busca em espaços já explorados.

Importante ressaltar que os espaços complementares vão se acumulando durante o processo e que os ramos à esquerda são resolvidos de maneira exata.

FIGURA 5.1 - PROCESSO DO IRSS



FONTE: Adaptado de Pécora Jr. (2008).

A resolução exata que acontece nos braços à esquerda da árvore pode ser feita por um *sol/ver* e resultar em 4 possíveis tipos de solução: (1) ótima, a melhor solução possível; (2) factível, uma solução viável qualquer sem provas de que é ótima; (3) desconhecida, nenhuma solução viável encontrada; ou (4) infactível, não existe solução para o problema. Essas respostas podem ser usadas para caracterizar a natureza do IRSS como sendo exato ou heurístico. Quando a solução encontrada é ótima ou infactível significa que o Espaço Restrito foi completamente explorado e não há mais nada a ser feito no ramo. Agora se o retorno do método exato é uma solução factível ou desconhecida não pode ser descartada a hipótese de que ainda haja melhores soluções dentro daquele RS e que ele não foi completamente explorado.

O *local branching* (FISCHETTI; LODI, 2003) é um algoritmo que, semelhante ao IRSS, explora espaços usando algum algoritmo exato em uma estrutura de árvore binária. Ele foi desenvolvido inspirado nos algoritmos de busca local e

depende de soluções encontradas em uma vizinhança limitada para ter continuidade, ou seja, os ramos à esquerda têm de encontrar soluções cada vez melhores para gerar novas vizinhanças a serem exploradas nos ramos que virão à direita. O IRSS apresenta uma grande vantagem por conseguir diversificar mais os espaços explorados, fugindo de ótimos locais. No IRSS os espaços restritos são limitados de maneira independente com a utilização de algum algoritmo heurístico.

Métodos heurísticos não garantem a otimalidade de suas respostas, isso quer dizer que não há garantia alguma que a melhor solução seja alcançada. Pode ser feito com que o IRSS assuma um caráter heurístico desde que não seja obrigado a atingir valores ótimos na fase de busca local. Isso pode acontecer quando os RS não são suficientemente pequenos para serem explorados completamente e a resposta pode ser a melhor obtida após algum limite de tempo. O problema dessa abordagem é que se o espaço restrito que contém alguma solução ótima não a encontra, esse espaço jamais será explorado novamente, logo, essa solução dificilmente será encontrada.

“O método pode ser exato. Para isso acontecer, cada ramificação a esquerda, RS, deve ser completamente explorada e um lower bound (upper bound para problemas de maximização) precisa ser calculado para cada ramificação a direita com relaxação linear. Então, se esse bound for pior que a melhor solução encontrada até o momento, a busca é finalizada e a solução pode ser declarada ótimo global. Apesar da capacidade de provar a otimalidade, o número de ramificações necessita ser grande.” (PÉCORA JR, 2008, p. 96, tradução nossa)

Existe um dilema em definir alguns parâmetros do IRSS, como a quantidade de variáveis a serem deixadas livres ou fixadas. Quanto menos variáveis fixadas maior é o espaço restrito de busca, em um espaço maior, maiores são as chances de alcançar uma solução melhor, porém maior será o tempo de resolução. Já quanto mais variáveis são fixadas, menor o espaço restrito e mais rápido é sua exploração, em compensação a qualidade da solução depende da qualidade do método heurístico e das variáveis escolhidas na fase de restrição.

5.2 APLICAÇÃO DO IRSS E RESULTADOS

Essa seção tem o objetivo de explicar alguns conceitos utilizados para aplicar a IRSS no PCARP-CM, sendo que a heurística responsável por auxiliar na

criação dos espaços restritos foi o ACO. O detalhamento de uma aplicação sendo exata e outra heurística e seus respectivos resultados também são apresentados e comparados com os melhores resultados obtidos com o novo modelo matemático.

5.2.1 Uso do IRSS no PCARP-CM

Para a aplicação do IRSS dentro do PCARP-CM é necessário identificar as variáveis mais relevantes do problema, ou seja, suas variáveis estruturais. As variáveis y_i parecem uma boa opção, pois indicam os pontos de saída e retorno de cada carro, dessa forma seria feita uma interessante limitação dos espaços. Porém, essas variáveis y_i são compostas pelo resultado da escolha dos caminhos no primeiro e último dia, sendo uma soma das outras variáveis de deslocamento que assumem valor um nesses períodos; vale ainda ressaltar que elas não existiam na modelagem antiga. Ademais, a quantidade de variáveis y_i no modelo é extremamente baixa comparada com as demais, se somente elas fixadas, o espaço restrito permanece grande e difícil de ser explorado. Definir o ponto de início e fim da rota no modelo também não ajuda a evitar soluções simétricas, pois qualquer ponto é candidato e o intervalo de tempo entre essas visitas pode acontecer diversas vezes.

As variáveis escolhidas como estruturais do modelo são as que representam deslocamentos x_{ijkp} , sendo elas triviais na formação dos espaços restritos. Quando fixados alguns deslocamentos em alguns dias, a resposta do modelo deve se ajustar completamente. Isso pode até mesmo diminuir ou restringir as opções de pontos de partida para outros carros. O objetivo prioritário do modelo é evitar atrasos em atendimentos e isso só acontece quando os arcos são atendidos nos dias corretos. Além disso, a solução retornada pelo algoritmo ACO é composta simplesmente por deslocamentos. Por esses motivos, a variável estrutural escolhida representa os deslocamentos.

Será aqui comparada uma abordagem exata e outra heurística do método IRSS. Ambas utilizam espaços restritos de tamanhos que podem se modificar a cada iteração, a exata utiliza espaços que são completamente explorados e a heurística espaços explorados enquanto não for atingido algum critério de parada. Detalhes dos processos serão expostos a seguir.

5.2.2 A fase de restrição

A criação de um RS está vinculada à inserção de uma restrição no modelo e, para definir essa restrição, informações são necessárias. A utilização de algoritmos construtivos, redução ou relaxação do problema principal ou outras heurísticas podem ser usadas para esse fim. Contudo, para um bom desempenho do método IRSS é necessário um bom gerador de soluções iniciais e uma alternativa pode ser o ACO.

A fase de restrição neste trabalho é guiada pelo Algoritmo ACO descrito no Capítulo 4, o algoritmo conta com m formigas cada uma gerando uma rota em cada iteração. Após gerada cada rota são aplicados operadores de busca local que melhoram a rota, então cada formiga despeja no caminho usado uma quantidade de feromônios que depende da qualidade da solução. Muitas soluções são avaliadas com esse algoritmo que, a princípio, retorna apenas a melhor solução. Durante o processo de otimização, os feromônios são acumulados em uma matriz que se atualiza a cada iteração, ela guarda o histórico de utilização dos arcos em cada dia. As informações dessa matriz de feromônios são valiosas, pois mostram quais arcos foram os mais usados e podem ser de grande auxílio na limitação de uma região restrita.

A matriz de feromônios passa a ser também um retorno do algoritmo ACO para ser usado na IRSS, resta extrair arcos relevantes dos dados disponibilizados usando algum método de classificação. Na gestão de materiais é usual classificar quais materiais são mais importantes em um inventário com uma técnica chamada classificação ABC seguindo o princípio de Pareto (JURAN; GODFREY, 1998). Tem-se a ideia de que uma porcentagem pequena de materiais pode representar uma grande porcentagem do consumo de clientes, esses podem ser chamados de Classe A, usualmente 20% dos itens representam aproximadamente 70% de vendas ou consumo.

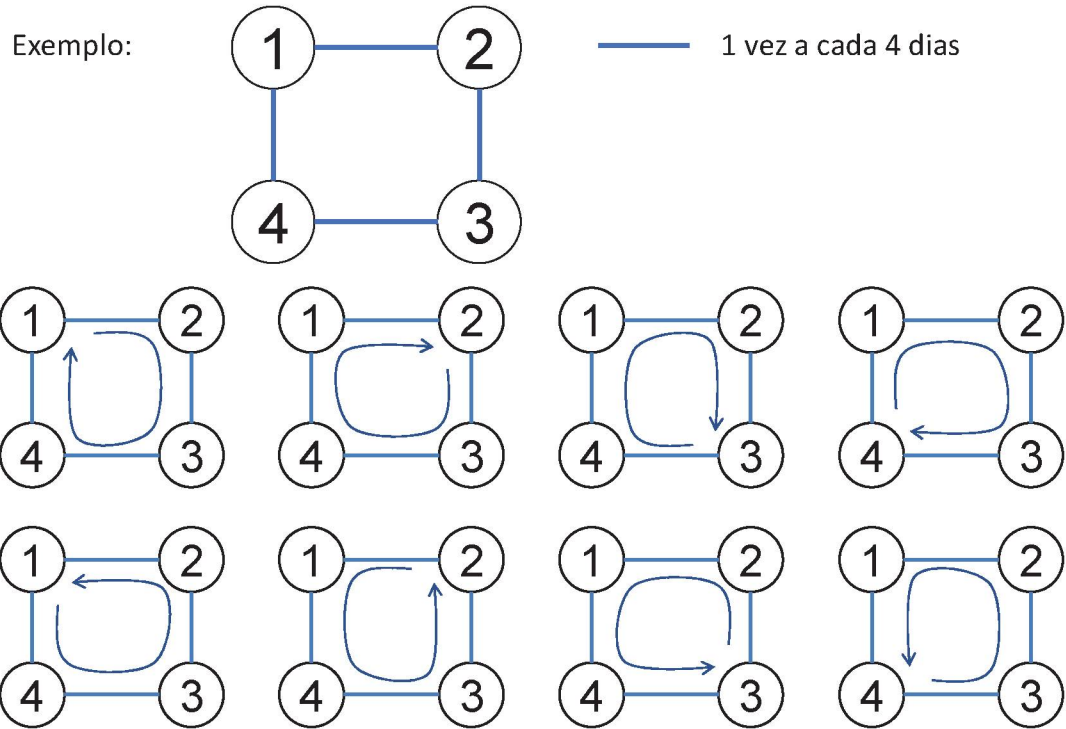
Cada arco presente na matriz de feromônios tem uma quantidade diferente de feromônios diária por carro, quanto mais usado, mais feromônio presente. Para auxiliar na limitação dos espaços restritos pode-se classificar alguns arcos como sendo de Classe A em cada dia de planejamento e cada carro. O processo segue da seguinte forma, inicialmente os arcos são ordenados segundo a quantidade de feromônios que receberam do maior para menor. A quantidade total de feromônio

despejado no dia é contabilizada e usada para calcular a proporção recebida por cada arco. Os arcos de Classe A serão os primeiros que formarem uma porcentagem acumulada maior ou igual a 70%. Como a solução final dada pelo algoritmo ACO é a melhor encontrada até então, todos os arcos usados na sua composição também são classificados com Classe A se ainda não o foram. Isso quer dizer que por mais que os arcos da solução não tenham recebido uma quantidade relevante de feromônios, serão classificados como A por representarem a melhor solução conhecida.

A definição de quantas e quais variáveis serão fixadas para restringir um espaço de busca serão diferentes para as versões exata e heurística e detalhadas mais à frente. É importante lembrar que ao limitar uma região com uma sequência de visitas, é necessário conhecer seu complementar. Uma mesma solução pode ser representada de várias maneiras (soluções simétricas) em virtude da ciclicidade do problema. Deste modo, se um pedaço de solução for usado para restringir uma região pode-se tirar vantagem dele para evitar seus simétricos. Essa situação será ilustrada com o exemplo da FIGURA 5.2 que apresenta 8 soluções simétricas.

Na FIGURA 5.2 tem-se 8 soluções simétricas, nas quais observa-se que a sequência de visitas é sempre a mesma, porém começando em pontos diferentes, podendo ser eles o 1, 2, 3 ou 4 e com sentidos diferentes, horário ou anti-horário. Se ao limitar o espaço solução desse problema ocorresse a imposição de que o carro faça o movimento 1-2 no primeiro dia e 2-3 no segundo dia, apenas uma solução é viável, sendo ela 1-2-3-4-1. Se um espaço complementar fosse criado de forma a evitar apenas que 1-2 aconteça no primeiro dia e 2-3 no segundo dia, isso ainda permitiria encontrar todas as outras 7 soluções, porém são iguais. Isso mostra que se tem a oportunidade de restringir mais o espaço complementar evitando que a sequência 1-2-3 apareça em qualquer situação já que ela não gerará nada melhor do que o que foi encontrado. No caso da FIGURA 5.2, evitar a sequência 1-2-3 partindo de qualquer ponto e em qualquer sentido tornaria o problema infactível, mas a melhor resposta com essa sequência já teria sido avaliada e guardada. O mesmo valeria se fosse fixado o movimento 1-2 no dia um e o movimento 3-4 no dia 3, o importante é saber o espaçamento entre as fixações.

FIGURA 5.2 - UM PEQUENO PROBLEMA COM TODAS SUAS SOLUÇÕES SIMÉTRICAS



FONTE: o autor (2018)

Para criação do RS complementar foi adotada a estratégia ilustrada, porém expandida para múltiplos carros. Seja VF o conjunto das variáveis x_{ijkp} que assumem valor 1 e foram fixadas para restringir o espaço solução formando um RS por meio da equação (23).

$$\sum_{x_{ijkp} \in VF} (1 - x_{ijkp}) = 0 \quad (23)$$

Já que os carros são homogêneos, o complementar deve excluir toda permutação de carros que usariam a mesma sequência de atendimentos. Por exemplo, em um problema com 3 carros, se trocados os índices dos carros de uma solução ótima ela continuará retornando o mesmo valor e a mesma sequência de atendimentos. O espaço complementar usará PC como o conjunto que contém todas as permutações $prmt$ de carros, cada $prmt \in PC$ é composto pela sequência de índices dos carros. Seja $Posi(k, prmt)$ o índice do carro na posição k da $prmt$, por exemplo, um problema com 3 carros tem $PC = \{(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)\}$, então $Posi(1,1) = 1$, pois $prmt = 1$ é dada por $(1,2,3)$, já

$Posi(2,4) = 3$, pois $prmt = 4$ é $(2,3,1)$. O espaço complementar ao RS criado em (23) pode ser expresso por (24).

$$\sum_{x_{ijkp} \in VF} (1 - x_{ij,Posi(k),f(p+d)}) \geq 1 \quad \forall prmt \in PC, \forall d \in H \text{ e} \quad (24)$$

$$f(r) = r, \text{ se } r \leq np \text{ ou } f(r) = (r) \bmod(np), \text{ se } r > np$$

O início dos atendimentos em pontos diferentes pode conduzir a soluções simétricas conforme mostrado na explicação da FIGURA 5.2. O ponto de início pode ser alterado apenas com a mudança do dia dos atendimentos de cada arco. A característica que mantém uma solução igual a outra, simétrica, é o intervalo de tempo entre os atendimentos serem iguais prevalecidos os sentidos. Então a equação (24) deve valer para qualquer dia desde que mantidas as distâncias temporais entre atendimentos, a ideia usada na equação (12) para os períodos que excedem o horizonte de tempo np aqui também deve ser usada. A equação (24) garante apenas um dos sentidos, o sentido contrário pode ser dado pela equação (25):

$$\sum_{x_{ijkp} \in VF} (1 - x_{ji,Posi(k),f(np+1-p+d)}) \geq 1 \quad \forall prmt \in PC, \forall d \in H \text{ e} \quad (25)$$

$$f(r) = r, \text{ se } r \leq np \text{ ou } f(r) = (r) \bmod(np), \text{ se } r > np$$

A FIGURA 5.3 mostra um exemplo com soluções simétricas que podem ser evitadas usando as equações (24) e (25). Supondo um problema, como o *pgdbcm6*, com 3 carros, um horizonte de planejamento de 14 dias e uma solução conhecida. Alguns arcos, como os de Classe A, podem ser utilizados para restringir um espaço e no caso da figura são os com destaque em amarelo formando o conjunto VF . Observa-se que existe a simetria quando são alterados os índices dos carros, os pontos de início e o sentido dos trechos. As próximas buscas podem evitar encontrar as soluções que alternam o índices dos carros quando se faz com que as equações (24) e (25) variem $\forall prmt \in PC$ e as variáveis x_{ijkp} tenham seus índices k convertidos pela função $Posi(k, prmt)$. A alteração dos inícios/partida dos carros e que mantém a distância, em tempo, entre atendimentos da Classe A é evitado ao

fazer as equações valerem para todo dia pertencente ao horizonte de tempo, $\forall d \in H$, e a variável x_{ijkp} se torna $x_{ijk,f(p+d)}$ com $f(r) = r$ se $r \leq np$ ou $f(r) = (r) \bmod (np)$ se $r > np$. A alteração dos sentidos é evitada quando a variável x_{ijkp} muda para $x_{jik,np+1-p}$, nesse caso, todas as características acumuladas usariam $x_{ji,Posi(k),f(np+1-p+d)}$.

FIGURA 5.3 - EXEMPLOS DE SOLUÇÕES SIMÉTRICAS QUE PODEM SER EVITADAS EM UM PROBLEMA COM 3 CARROS

Solução														
Carro	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14
1	2	4	3	11	12	5	3	2	1	12	6	5	11	4
2	1	2	9	10	8	7	7	7	8	10	1	12	7	6
3	7	8	10	1	12	7	1	4	4	3	11	12	5	3
Solução	456													
Exemplo simétrico alterando carros a ser evitado														
Carro	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14
3	2	4	3	11	12	5	3	2	1	12	6	5	11	4
1	1	2	9	10	8	7	7	7	8	10	1	12	7	6
2	7	8	10	1	12	7	1	4	4	3	11	12	5	3
Solução	456													
Exemplo simétrico alterando inícios e carros a ser evitado														
Carro	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14
2	3	2	4	3	11	12	5	3	2	1	12	6	5	11
3	4	1	2	9	10	8	7	7	7	8	10	1	12	7
1	6	7	8	10	1	12	7	1	4	4	3	11	12	5
Solução	456													
Exemplo simétrico alterando inícios, carros e sentidos a ser evitado														
Carro	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14
1	4	11	5	6	12	1	2	3	5	12	11	3	4	2
3	6	7	12	1	10	8	7	7	7	8	10	9	2	1
2	3	5	12	11	3	4	4	1	7	12	1	10	8	7
Solução	456													

FONTE: o autor (2018)

Na FIGURA 5.3 é importante ressaltar novamente que as variáveis que ficariam fixadas seriam apenas as em amarelo e a melhor solução a ser encontrada seria obtida com a variação das demais. Evitar soluções simétricas é de extrema importância para a resolução do modelo, de maneira que elas não serão avaliadas em etapas futuras. O número de simetrias cresce muito nesse problema em virtude da ciclicidade e a não definição de um ponto de partida obrigatório, além disso a quantidade de soluções é muito grande. Essa estratégia pode ajudar a provar a otimalidade de algum problema.

5.2.3 A fase de busca

A fase de busca é a simples resolução do modelo com RS utilizando um método exato. No caso aqui abordado será utilizado o Gurobi 8.0.1 e alguns ajustes podem depender da forma com que o IRSS será usado. Se for exato, é obrigatório que o método alcance soluções ótimas em cada etapa, então uma estratégia deve ser definida para isso. Já se o método for heurístico, devem ser estabelecidos critérios de parada durante a resolução, podendo ser um limite de tempo total e/ou baseado no monitoramento do desempenho do método, por exemplo, tempo sem melhoria.

5.2.4 Estratégia IRSS Exata

Essa abordagem necessita garantir que cada RS seja explorado completamente, porém como isso pode ser feito? Vinda a melhor solução conhecida do ACO, alguns movimentos devem ser escolhidos para serem obrigatórios restringindo o espaço solução. Se fixados 100% dos movimentos, o *solver* não conseguirá otimizar em nada o modelo, se 0% dos movimentos forem fixados o *solver* terá o espaço solução completo para ser explorado. Torna-se necessário determinar uma porcentagem intermediária que permita que o *solver* encontre valores ótimos.

A porcentagem de movimentos a serem fixados varia dependendo da qualidade dos movimentos escolhidos, as vezes a escolha de uma quantidade pequena de movimentos importantes pode levar a uma exploração mais rápida do que a fixação de movimentos menos relevantes. Por exemplo, se o movimento escolhido é para um arco cuja periodicidade é igual ao horizonte de tempo, ele tem necessidade única, logo menos simetrias existirão. Já a fixação de um arco que tem mais frequências não indica qual é aquela necessidade atendida. Não necessariamente a periodicidade dita a importância do movimento fixado, mas uma combinação de outros fatores como a estrutura do grafo, os graus dos nós que o compõe o arco e as necessidades dos arcos adjacentes. O importante é conseguir variar a porcentagem de variáveis fixadas para alcançar uma resposta ótima pelo *solver*.

Os RS devem ser pequenos o suficiente para serem completamente explorados, quanto mais espaços explorados maior a chance de encontrar alguma solução muito boa. Para essa estratégia estipulou-se que 10 minutos devem ser o tempo máximo para encontrar a solução ótima em um espaço, isso acontecendo significa que ele foi suficientemente pequeno para ser explorado por completo. Logo, foi decidido que a quantidade de movimentos a serem fixados para gerar esses espaços poderia variar entre as opções de um vetor $Porcent = [5\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%]$. Essas porcentagens são testadas gradualmente e nesses testes é possível a obtenção de soluções incumbentes, quanto menor a probabilidade mais liberdade o modelo tem, porém menores são as chances de provar valores ótimos.

Dada uma porcentagem faz-se necessário determinar quais as variáveis da melhor solução obtida com o ACO serão fixadas. Tendo a informação da matriz de feromônios, pode ser contado quantos arcos de Classe A pertencem a cada dia e a cada carro. Os dias e carros que tem uma menor quantidade de arcos Classe A são interessantes, pois esses arcos receberam quantidades relevantes de feromônio, ou seja, foram muitas vezes avaliados e usados. Sendo assim, serão escolhidos os movimentos da melhor solução vinda do ACO, que estão relacionados aos dias e carros com menores quantidades de arcos Classe A até atingir a porcentagem desejada.

Crítérios de parada devem ser estabelecidos para que o algoritmo finalize. Até o momento foi utilizado tempo limite de 24 horas ao longo desse trabalho, então optou-se por continuar com esse tempo limite para uma comparação justa. Ademais, o algoritmo também será parado quando 5 espaços restritos explorados por completo e consecutivamente não retornarem no mínimo valor igual a melhor solução ótima conhecida. Podem existir múltiplas soluções com mesmos valores além das simétricas, elas não contabilizam para o critério de parada, pois podem compor um espaço complementar cada vez menor.

O ALGORITMO 2 representa o algoritmo IRSS aplicado de forma exata. Inicialmente o modelo matemático será gerado para o problema e serão estabelecidos os critérios de parada. Então, o algoritmo ACO (ALGORITMO 1) é usado para auxiliar na fase de restrição, na qual uma quantidade de variáveis será fixada segundo uma porcentagem. É dado um tempo de 10 minutos para o ACO, podendo ele ser interrompido dentro da fase de melhoria. Cada porcentagem é

testada de forma crescente até que algum espaço restrito seja completamente explorado na fase de busca, ou seja, o *solver* retorna à solução ótima ou mostra que o modelo é infactível. As melhores soluções encontradas durante os testes são armazenadas, pois não há motivos para descartá-las. Os critérios de parada são avaliados após cada espaço restrito explorado por completo. Caso reste tempo para completar as 24 horas limite, o espaço complementar passa a ser avaliado até atingir o limite de tempo total, caso contrário, ele é avaliado por 10 minutos na esperança de provar otimalidade com o *lower bound* obtido.

ALGORITMO 2 - IRSS EXATO

1:	procedimento IRSS Exato (carros, arcos, cidades)
2:	Gerar o modelo matemático para o problema e estabelecer critérios de parada
3:	Faça até alcançar um critério de parada
4:	Gere soluções utilizando o ACO (ALGORITMO 1) com um tempo limite de 10 minutos
5:	Para cada porcentagem $// (i = 1 \text{ to } m)$
6:	Determine os <i>Percent[i]</i> de variáveis a serem fixadas
7:	Crie o RS adicionando a restrição com as variáveis determinadas
8:	Resolva o modelo com tempo máximo de 10 minutos usando a como solução inicial a obtida do ACO
9:	Remova as restrições que geraram o RS
10:	Se a solução obtida é a melhor conhecida então a salve como sendo a melhor
11:	Se a solução obtida é ótima então adicione todas as restrições do RS complementar, salve a solução encontrada e saia do Para
12:	Fim Faça
13:	Se tempo transcorrido até o momento é menor que 24 horas então resolva o modelo com as restrições complementares acumuladas até obter uma solução ótima ou atingir um tempo total acumulado de 24 horas
14:	Se tempo transcorrido até o momento é maior ou igual a 24 horas então resolva o modelo com as restrições complementares por 10 minutos
15:	Se o <i>lower bound</i> do modelo complementar é maior que a melhor solução conhecida então o ótimo foi encontrado
16:	Retorne a melhor solução conhecida

FONTE: O autor (2018).

As respostas obtidas para as 39 instâncias *pgdbcm* estão à disposição na TABELA 5.1.

TABELA 5.1 - RESPOSTAS DO IRSS EXATO PARA OS 39 *pgdbcm*

Instâncias	IRSS - Espaços Restritos						IRSS - Complementar			IRSS Geral		
	Tempo (seg)	Melhor	Iteração	Exato	Iteração	Crit. Para.	Tempo (seg)	L. Bound	Solução	Ótimo	Solução	Tempo Acum.
<i>pgdbcm1</i>	17704	451	4	451	4	1	67015	456	456	Sim	451	84720
<i>pgdbcm2</i>	13555	604	6	608	1	1	72844	511	6580	Não	604	86399
<i>pgdbcm3</i>	86401	340	1	340	46	2	600	324	1366	Não	340	87001
<i>pgdbcm4</i>	31523	1560	9	1560	10	1	54876	447	3513	Não	1560	86399
<i>pgdbcm5</i>	14527	2768	1	2768	1	1	71872	659	7755	Não	2768	86400
<i>pgdbcm6</i>	87587	456	1	456	45	2	600	418	467	Não	456	88187
<i>pgdbcm7</i>	20895	508	2	526	4	1	65505	494	516	Não	508	86400
<i>pgdbcm8</i>	17848	11385	6	16385	2	1	68552	326	6363	Não	6363	86400
<i>pgdbcm9</i>	17554	28452	1	28452	1	1	68846	439	20459	Não	20459	86401
<i>pgdbcm10</i>	32325	1798	9	3803	7	1	54075	684	751	Não	751	86401
<i>pgdbcm11</i>	22344	442	3	442	3	1	64058	413	448	Não	442	86403
<i>pgdbcm12</i>	20401	527	4	527	6	1	65998	471	536	Não	527	86400
<i>pgdbcm13</i>	30174	620	35	620	35	1	39	620	620	Sim	620	30214
<i>pgdbcm14</i>	6164	145	5	145	5	1	10	145	145	Sim	145	6175
<i>pgdbcm15</i>	24827	140	7	140	7	1	61572	137	140	Não	140	86399
<i>pgdbcm16</i>	32549	182	1	182	10	1	53854	176	182	Não	182	86403
<i>pgdbcm17</i>	17399	168	2	168	2	1	69000	161	168	Não	168	86400
<i>pgdbcm18</i>	44788	385	8	395	8	1	41611	369	377	Não	377	86401
<i>pgdbcm19</i>	53940	125	32	125	32	1	32459	122	125	Não	125	86400
<i>pgdbcm20</i>	27307	261	2	2277	6	1	59094	241	270	Não	261	86402
<i>pgdbcm21</i>	45982	212	19	212	19	1	40420	204	212	Não	212	86406
<i>pgdbcm22</i>	36074	427	2	448	5	1	50328	423	435	Não	427	86412
<i>pgdbcm23</i>	44981	336	2	341	2	1	41424	329	336	Não	336	86429
<i>pgdbcm24</i>	56171	574	1	574	6	1	30229	511	590	Não	574	86401
<i>pgdbcm25</i>	87400	340	13	340	19	2	602	315	367	Não	340	88004
<i>pgdbcm26</i>	41409	508	4	508	4	1	44990	447	530	Não	508	86399
<i>pgdbcm27</i>	41721	677	14	699	14	1	44679	639	676	Não	676	86400
<i>pgdbcm28</i>	45139	444	17	444	15	1	41261	411	444	Não	444	86400
<i>pgdbcm29</i>	21219	349	6	372	3	1	65181	305	345	Não	345	86400
<i>pgdbcm30</i>	54682	17439	11	20458	15	1	31717	378	44429	Não	17439	86401
<i>pgdbcm31</i>	18805	745	2	2935	2	1	67594	684	743	Não	743	86400
<i>pgdbcm32</i>	14606	523	3	523	3	1	71794	479	547	Não	523	86400
<i>pgdbcm33</i>	26696	157	13	157	13	1	59703	154	157	Não	157	86400
<i>pgdbcm34</i>	17519	120	1	120	5	1	68880	119	120	Não	120	86399
<i>pgdbcm35</i>	22241	170	3	177	4	1	64159	167	170	Não	170	86401
<i>pgdbcm36</i>	40073	385	1	435	6	1	46326	369	379	Não	379	86400
<i>pgdbcm37</i>	18880	264	1	1272	3	1	67519	246	264	Não	264	86400
<i>pgdbcm38</i>	42330	428	4	510	6	1	44070	423	426	Não	426	86407
<i>pgdbcm39</i>	23688	336	1	352	1	1	62712	329	336	Não	336	86418

FONTE: o autor (2018)

Na TABELA 5.1 tem-se 3 informações principais referentes às instâncias que estão na primeira coluna. O primeiro conjunto de informações diz respeito a exploração dos espaços restritos, sendo que o primeiro dado mostra o tempo decorrido até o critério de parada ser satisfeito. Em seguida tem-se o melhor resultado encontrado, lembrando que não necessariamente ele foi um resultado ótimo; logo ao lado tem-se a iteração em que foi encontrado. Os resultados ótimos estão à disposição na próxima coluna (Exato) e ao lado se encontra a respectiva iteração na qual ele foi encontrado. A última informação referente a essa etapa diz respeito ao critério de parada, em que (1) representa que o número máximo de iterações sem melhoria foi alcançado e (2) que o tempo limite foi atingido. Na sequência tem-se as informações obtidas quando o modelo tentou resolver a região complementar, são elas o tempo, o *lower bound* e a solução encontrada. E por fim, um resumo se o ótimo foi provado, qual a solução final e o tempo acumulado.

Foi possível provar a otimalidade de apenas 3 dos problemas. De maneira geral, a exploração dos espaços restritos alcançou as melhores soluções em 30 vezes enquanto a exploração do espaço complementar em 23 vezes. Vale ressaltar que houve 14 empates, pois foi possível alcançar soluções diferentes e de mesmo valor nas duas situações. O próximo capítulo faz uma comparação desse método com as soluções obtidas pelo *so/ver* puro e o *so/ver* usando solução inicial vinda do ACO (TABELA 4.4). É possível adiantar que a solução foi melhor ou igual em 29 das 39 instâncias.

Muitas informações podem ser obtidas do processo como um todo, detalhes da exploração nos espaços restritos podem ser encontrados na TABELA 5.2 e o GRÁFICO 5.1. Na tabela é apresentada a quantidade total de espaços explorados completamente (sem contabilizar o espaço complementar). A terceira coluna mostra quais porcentagens tiveram exploração completa concluída e a quantidade de vezes que foram usadas. Finalmente, a coluna 4 mostra a porcentagem usada que pode retornar a melhor solução, nota-se que elas são mais baixas, porém 10 minutos não é tempo suficiente para ser provado otimalidade.

TABELA 5.2 - DETALHES DA EXPLORAÇÃO DOS RS PARA AS 39 INSTÂNCIAS

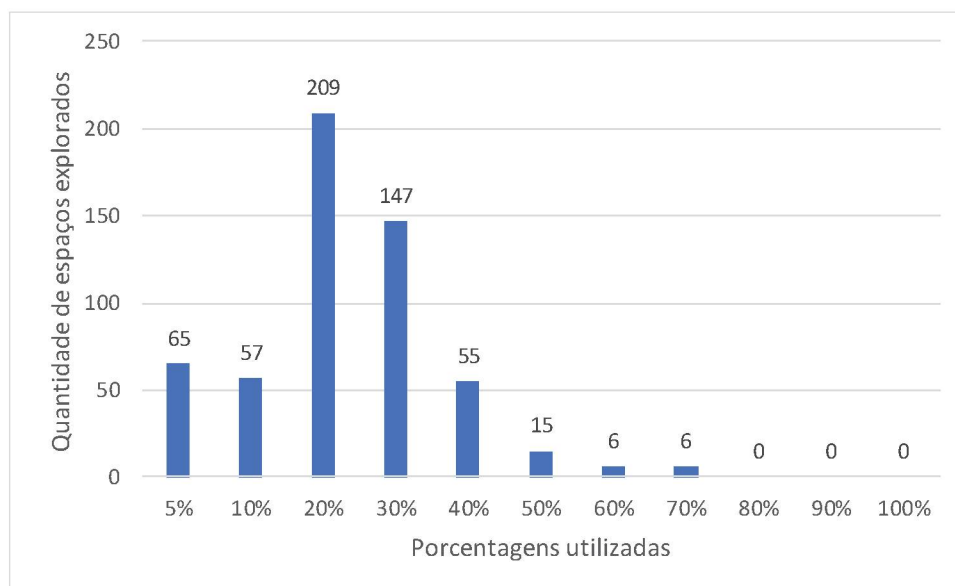
Instâncias	Quantidade de Espaços Explorados	Porcentagens usadas (quantidade)	Melhor valor encontrado com
<i>pgdbcm1</i>	9	20%(8);30%(1);	5%
<i>pgdbcm2</i>	6	20%(2);30%(4);	10%
<i>pgdbcm3</i>	46	5%(1);10%(6);20%(33);30%(6);	5%
<i>pgdbcm4</i>	15	10%(1);20%(8);30%(6);	20%
<i>pgdbcm5</i>	6	20%(2);30%(3);40%(1);	20%
<i>pgdbcm6</i>	45	10%(8);20%(29);30%(8);	5%
<i>pgdbcm7</i>	9	20%(4);30%(4);40%(1);	5%
<i>pgdbcm8</i>	7	20%(1);30%(5);40%(1);	20%
<i>pgdbcm9</i>	6	30%(2);40%(4);	30%
<i>pgdbcm10</i>	12	30%(8);40%(4);	10%
<i>pgdbcm11</i>	8	20%(1);30%(3);40%(3);50%(1);	5%
<i>pgdbcm12</i>	11	10%(2);20%(9);	5%
<i>pgdbcm13</i>	40	5%(36);10%(3);20%(1);	5%
<i>pgdbcm14</i>	10	5%(10);	5%
<i>pgdbcm15</i>	12	20%(9);30%(3);	20%
<i>pgdbcm16</i>	15	10%(2);20%(4);30%(9);	5%
<i>pgdbcm17</i>	7	20%(3);30%(2);40%(2);	5%
<i>pgdbcm18</i>	13	30%(2);40%(6);50%(4);60%(1);	20%
<i>pgdbcm19</i>	37	5%(13);10%(9);20%(13);30%(2);	10%
<i>pgdbcm20</i>	11	30%(10);40%(1);	5%
<i>pgdbcm21</i>	24	5%(3);10%(4);20%(8);30%(8);40%(1);	10%
<i>pgdbcm22</i>	10	30%(4);40%(2);50%(2);70%(2);	10%
<i>pgdbcm23</i>	7	40%(4);50%(3);	20%
<i>pgdbcm24</i>	11	20%(8);30%(3);	10%
<i>pgdbcm25</i>	19	10%(7);20%(12);	5%
<i>pgdbcm26</i>	9	5%(1);10%(2);20%(6);	5%
<i>pgdbcm27</i>	19	20%(10);30%(9);	10%
<i>pgdbcm28</i>	20	20%(8);30%(12);	10%
<i>pgdbcm29</i>	8	20%(1);30%(6);40%(1);	10%
<i>pgdbcm30</i>	20	30%(13);40%(7);	20%
<i>pgdbcm31</i>	7	30%(4);40%(3);	10%
<i>pgdbcm32</i>	8	10%(1);20%(7);	5%
<i>pgdbcm33</i>	18	5%(1);10%(10);20%(7);	10%
<i>pgdbcm34</i>	10	10%(2);20%(8);	5%
<i>pgdbcm35</i>	9	20%(4);30%(2);40%(3);	5%
<i>pgdbcm36</i>	11	30%(2);40%(5);50%(2);60%(1);70%(1);	10%
<i>pgdbcm37</i>	8	20%(3);30%(5);	5%
<i>pgdbcm38</i>	11	30%(1);40%(4);50%(2);60%(2);70%(2);	10%
<i>pgdbcm39</i>	6	40%(2);50%(1);60%(2);70%(1);	5%

FONTE: o autor (2018)

Em geral, os melhores valores encontrados, sem provar otimalidade, precisam de uma pequena quantidade de variáveis fixadas. Ao usar uma fixação máxima de 30% em uma das instâncias, a média de fixação ficou em 10%. O

GRÁFICO 5.1 mostra que a porcentagem utilizada que mais retornou valores ótimos foi 20%, seguida por 30% e 5%. Em nenhum momento foi necessário fixar mais de 80% das variáveis para encontrar alguma solução, em poucas vezes foi necessário fixar entre 50% e 70%.

GRÁFICO 5.1 - PORCENTAGENS UTILIZADAS NO IRSS QUE RETORNARAM VALOR ÓTIMO



FONTE: o autor (2018)

As variáveis que mais foram fixadas estavam em geral associadas aos primeiros dias com primeiro carro e aos últimos dias com último carro em virtude da natureza dos grafos criados para a movimentação das formigas e o despejar de feromônios. A fixação desses movimentos ajuda muito na obtenção de ótimos nos espaços restritos. Evitar soluções simétricas tem grande efetividade, mas mesmo assim existem muitas soluções que assumem o mesmo valor, como quando é possível que o carro folgue.

O algoritmo aceitou o critério de parada em algumas instâncias por causa de erros de arredondamento. Em dado momento veio a acontecer que algumas variáveis binárias que deveriam assumir o valor 1 acabaram assumindo um valor muito próximo, como, por exemplo, 0,999 e levaram a uma solução com valor pouco mais baixo que o valor inteiro exato. Quando isso ocorreu, muito provavelmente algumas funções de mesmo valor deixaram de ajudar na criação do espaço complementar final, mas mesmo assim o desempenho do algoritmo não foi afetado. É o caso da instância *pgdbcm19* que na iteração 32 alcançou 124,9999999064279 e na iteração 34 e 36 alcançou o valor 125, sendo finalizada na iteração 37. Da

TABELA 4.4 sabe-se que o valor é ótimo, porém aqui o tempo limite foi atingido sem fazer a prova e não se sabe quantas soluções que resultam em 125 existem. Nesse intervalo esse valor foi alcançado 15 vezes, 16 se alcançado o retorno do espaço complementar.

Outro fato que interfere muito na resolução é o desempenho da ACO. Por exemplo, a instância *pgdbcm6* teve o melhor resultado, 456, já na primeira iteração e mesmo assim houve outras 44 iterações. A fixação de 20%, mais frequente na instância, usa apenas 9 variáveis, que geralmente estão associadas aos movimentos nos primeiros dias de cada carro. Já que o ACO gera um ponto aleatório inicial cada vez que é usado, novas soluções variadas e boas são sempre geradas, conduzindo a valores baixos. A combinação dos dois algoritmos tem uma relação consistente, ambos têm um desempenho muito bom alcançando diversas vezes os mesmos baixos valores, que resultaram em sequências de atendimentos diversificadas.

5.2.5 Estratégia IRSS Heurística

Na abordagem heurística foi concedida mais liberdade ao modelo matemático, que não obrigatoriamente precisa retornar valores ótimos. Tanto a fase de restrição como a fase de busca sofreram alterações, dessa vez o *sol/ver* recebeu espaços maiores de busca que foram limitados de uma forma diferente. O algoritmo que auxilia a fase de restrição continuou sendo o ACO e a formação dos espaços complementares permaneceu semelhante.

Todos os arcos de Classe A em cada dia e cada carro, resultado da classificação vinda da matriz de feromônios, pareceram ser interessantes para a resolução do problema. A abordagem heurística usou todos os arcos classificados como Classe A para os respectivos dias e carros selecionados para compor um RS. Fato que a torna diferente da abordagem exata que fixou apenas os arcos que pertenciam à solução obtida pelo ACO.

As restrições (23), (24) e (25) precisaram ser substituídas respectivamente por (26), (27) e (28) pois agora não são todas as variáveis selecionadas que devem assumir valor 1. Seja VFH o conjunto que contém de todas as variáveis x_{ijkp} de Classe A para seus respectivos dias e carros selecionados, é necessário que

somente uma variável por dia e carro assuma valor 1. Seja $nroDC$ o número de dias e carros diferentes no conjunto VFH , então:

$$\sum_{x_{ijkp} \in VFH} (1 - x_{ijkp}) = |VFH| - nroDC \quad (26)$$

$$\sum_{x_{ijkp} \in VFH} (1 - x_{ij,Posi(k),f(p+d)}) \geq |VFH| - nroDC + 1$$

$$\forall prmt \in PC, \forall d \in H \text{ e} \quad (27)$$

$$f(r) = r, \text{ se } r \leq np \text{ ou } f(r) = (r) \bmod(np), \text{ se } r > np$$

$$\sum_{x_{ijkp} \in VFH} (1 - x_{ji,Posi(k),f(np+1-p+d)}) \geq |VFH| - nroDC + 1$$

$$\forall prmt \in PC, \forall d \in H \text{ e} \quad (28)$$

$$f(r) = r, \text{ se } r \leq np \text{ ou } f(r) = (r) \bmod(np), \text{ se } r > np$$

Aqui também se fez necessário escolher uma porcentagem de dias e carros para terem suas variáveis fixadas. A quantidade selecionada de dias e carros continuou sendo composta pelos que contém menores quantidade de arcos Classe A até atingir a porcentagem estipulada. Sem a obrigação de encontrar o ótimo em cada espaço restrito, a porcentagem de fixação oscilou de iteração para iteração e não dentro de cada iteração como no IRSS exato. A oscilação seguiu as porcentagens do vetor $PercentH = [50\%, 40\%, 30\%, 20\%, 10\%]$, espaços cada vez maiores foram avaliados até atingir a solução ótima dentro de um tempo limite de 6 horas ou 2 horas sem melhorias; caso ocorressem mais de 5 iterações, as porcentagens voltariam a se repetir.

Espaços pequenos avaliados inicialmente geram um número mais limitado de soluções que serão evitadas nas próximas iterações. Conforme os espaços vão aumentando, pedaços menores do espaço solução já foram eliminados e as novas soluções que podem vir devem ser diferentes das já encontradas – lembrando que os espaços maiores contêm muito mais soluções. Ir de regiões menores para maiores é mais intuitivo nesse tipo de situação na qual não é necessário encontrar o valor ótimo, pois é possível que uma região maior contenha um espaço menor já explorado e não fará buscas nele. A ideia é encontrar soluções muito boas em

regiões pequenas e ir procurando outras melhores, podendo ser ao redor dessa região pequena ou não, mas que não sejam as mesmas.

Esse crescimento da região visa evitar o descarte de soluções muito boas, pois essa é uma desvantagem ao utilizar o algoritmo de forma heurística. A solução incumbente retornada depois de um tempo pode não ser a ótima e, em dado momento, a tarefa de encontrar a ótima pode se tornar impossível. Por exemplo, se um problema contém apenas uma rota ótima e ela pertencer a um espaço restrito que não foi explorado completamente, então ela não poderá ser encontrada em outros espaços.

ALGORITMO 3 - IRSS HEURÍSTICO

1:	procedimento IRSS Exato (carros, arcos, cidades)
2:	Gerar o modelo matemático para o problema e estabelecer critérios de parada, faça $i = 0$.
3:	Faça até alcançar um critério de parada
4:	Gere soluções utilizando o ACO (ALGORITMO 1) com um tempo limite de 10 minutos
5:	Determine os $PercentH[(i)mod(5) + 1]$ de variáveis que compõe <i>VFH</i>
6:	Crie o RS adicionando a restrição com as variáveis de <i>VFH</i>
7:	Resolva o modelo com tempo máximo de 6 horas usando a como solução inicial a obtida do ACO, caso o <i>solver</i> não apresente melhorias no período de duas horas também pare
8:	Remova as restrições que geraram o RS
9:	Se a solução obtida é a melhor conhecida então a salve como sendo a melhor
10:	Adicione as restrições que geram o espaço complementar
11:	$i \leftarrow i + 1$
12:	Fim Faça
13:	Se tempo transcorrido até o momento é menor que 24 horas então resolva o modelo com as restrições complementares acumuladas até obter uma solução ótima ou atingir um tempo total acumulado de 24 horas
14:	Se tempo transcorrido até o momento é maior ou igual a 24 horas então resolva o modelo com as restrições complementares por 10 minutos
15:	Se todas as RS exploradas retornaram valores ótimos e o <i>lower bound</i> do complementar é maior ou igual a melhor solução conhecida então o ótimo foi encontrado
16:	Retorne a melhor solução conhecida

FONTE: o autor (2018)

O IRSS Heurístico tem uma estrutura semelhante ao ALGORITMO 2 (Exato), as diferenças podem ser encontradas no ALGORITMO 3. O procedimento

inicia recebendo todos os dados, o modelo matemático é criado, é feito $i = 0$ para controlar as porcentagens utilizadas e são estabelecidos os critérios de parada. Como critério de parada é estabelecido o tempo limite também de 24 horas, dessa vez o IRSS poderá fazer 5 iterações consecutivas sem conseguir melhoria, ou seja, ao finalizar uma quinta exploração em que o resultado obtido não superou o resultado melhor a exploração de RS não continua. É importante notar que soluções de mesmo valor não são mais tão interessantes, pois não há a obrigação de provar otimalidade. O algoritmo ACO gera as informações necessárias para determinar as variáveis a serem fixadas, na fase de restrição os RS são criados com as variáveis de *VFH* e a fase de busca se inicia com os critérios estabelecidos. As restrições adicionadas para compor o RS são removidas. Se a solução obtida for a melhor, ela é salva, em seguida o espaço complementar é formado e o valor de i atualizado. O procedimento é feito até o critério de parada ser satisfeito de maneira que a fase de busca não é interrompida no meio. Finalmente, o espaço complementar é explorado a fim de encontrar melhores valores ou provar otimalidade se possível. O IRSS Heurístico retorna o melhor valor encontrado.

Os resultados alcançados com o IRSS Heurístico estão à disposição na TABELA 5.3. Novamente as informações estão dispostas em três blocos importantes representando o que foi alcançado com os espaços restritos, com o espaço complementar ao final e o resultado geral. Quanto aos espaços restritos tem-se as informações de tempo consumido, a solução encontrada e a respectiva iteração em que isso ocorreu, o critério de parada (1) representa que o número máximo de iterações sem melhoria foi alcançado, (2) o tempo limite foi atingido e (3) a memória do computador esgotou; finalmente tem-se um indicador se todo RS atingiu valor ótimo. Quanto aos espaços complementares tem-se o tempo gasto, o *lower bound* que faz sentido apenas se todos os espaços restritos explorados até o momento fossem completamente explorados e, finalmente, a solução encontrada. As últimas 3 colunas mostram se a solução encontrada foi ótima, qual a melhor solução e o tempo total acumulado.

É possível notar que a melhor solução encontrada ocorreu em algum espaço restrito em 38 das instâncias, exceto para a instância *pgdbcm31* que apresentou o melhor resultado no espaço complementar.

TABELA 5.3 - RESPOSTAS DO IRSS HEURÍSTICO PARA OS 39 *pgdbcm*

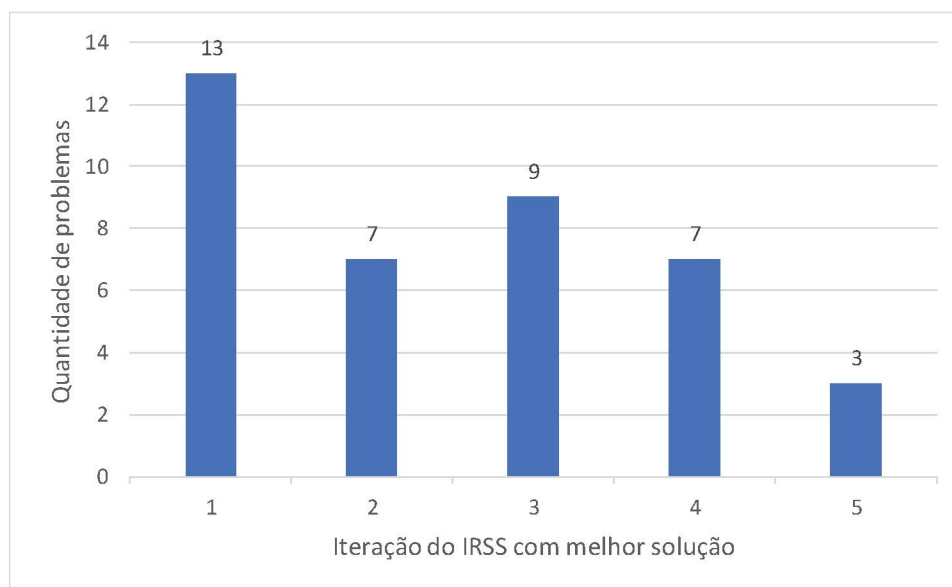
Instâncias	IRSS - Espaços Restritos					IRSS - Complementar			IRSS Geral		
	Tempo	Solução	Iteração	Critério de Parada	Todos ótimos	Tempo	Lower Bound	Solução	Ótimo	Solução	Tempo Acum.
<i>pgdbcm1</i>	6675	451	2	1	Não	69724	-	451	Não	451	86399
<i>pgdbcm2</i>	7998	576	4	2/1	Não	600	-	42573	Não	576	88598
<i>pgdbcm3</i>	5753	340	1	1	Sim	80646	318	340	Não	340	86400
<i>pgdbcm4</i>	51501	1560	2	1	Não	34899	-	4520	Não	1560	86401
<i>pgdbcm5</i>	83016	7757	4	1	Não	3384	-	13764	Não	7757	86401
<i>pgdbcm6</i>	5016	456	2	1	Sim	81383	418	456	Não	456	86399
<i>pgdbcm7</i>	89199	516	3	2/1	Não	600	-	3572	Não	516	89799
<i>pgdbcm8</i>	86988	26359	3	2/1	Não	600	-	1E+100	Não	26359	87599
<i>pgdbcm9</i>	79637	33472	1	1	Não	6762	-	192391	Não	33472	86439
<i>pgdbcm10</i>	87001	5744	2	2/1	Não	600	-	49740	Não	5744	87610
<i>pgdbcm11</i>	86702	442	5	2/1	Não	600	-	1E+100	Não	442	87304
<i>pgdbcm12</i>	19511	527	3	1	Não	66889	-	539	Não	527	86400
<i>pgdbcm13</i>	24863	620	1	1	Não	36	-	620	Não	620	24900
<i>pgdbcm14</i>	5193	145	3	1	Sim	7	145	145	Sim	145	5201
<i>pgdbcm15</i>	55191	140	3	1	Não	31208	-	140	Não	140	86400
<i>pgdbcm16</i>	48330	182	1	1	Não	38069	-	182	Não	182	86401
<i>pgdbcm17</i>	86441	168	3	2	Não	600	-	170	Não	168	87043
<i>pgdbcm18</i>	71583	369	1	1	Não	14816	-	4418	Não	369	86409
<i>pgdbcm19</i>	11284	125	2	1	Sim	2202	125	125	Não	125	13488
<i>pgdbcm20</i>	78296	2279	3	1	Não	8104	-	10273	Não	2279	86407
<i>pgdbcm21</i>	56237	212	1	1	Não	30164	-	212	Não	212	86414
<i>pgdbcm22</i>	20646	426	1	3	-	-	-	-	Não	426	-
<i>pgdbcm23</i>	81761	336	3	1	Não	4641	-	129432	Não	336	86450
<i>pgdbcm24</i>	88055	616	2	2/1	Não	600	-	8582	Não	616	88656
<i>pgdbcm25</i>	6423	340	4	1	Sim	79976	315	340	Não	340	86400
<i>pgdbcm26</i>	34730	519	5	1	Não	51669	-	522	Não	519	86399
<i>pgdbcm27</i>	87145	676	4	2	Não	600	-	8662	Não	676	87746
<i>pgdbcm28</i>	43071	444	1	1	Não	43328	-	444	Não	444	86400
<i>pgdbcm29</i>	64647	349	1	1	Não	21752	-	143297	Não	349	86409
<i>pgdbcm30</i>	85164	21455	2	1	Não	1236	-	1E+100	Não	21455	86445
<i>pgdbcm31</i>	72800	2852	1	1	Não	13599	-	755	Não	755	86413
<i>pgdbcm32</i>	6321	523	4	1	Sim	52473	523	523	Sim	523	58795
<i>pgdbcm33</i>	41621	157	1	1	Não	44779	-	157	Não	157	86400
<i>pgdbcm34</i>	32679	120	3	1	Não	53720	-	120	Não	120	86400
<i>pgdbcm35</i>	70866	170	5	1	Não	15534	-	170	Não	170	86401
<i>pgdbcm36</i>	87928	371	4	2	Não	600	-	82467	Não	371	88536
<i>pgdbcm37</i>	86755	264	4	2	Não	87359	-	264	Não	264	87359
<i>pgdbcm38</i>	61648	428	1	1	Não	-	-	-	Não	428	-
<i>pgdbcm39</i>	80343	339	1	1	Não	6064	-	29402	Não	339	86502

FONTE: o autor (2018)

Tanto a instância 38 como a instância 22 tiveram problemas com a falta de memória dos notebooks. Para exemplificar o ocorrido, o modelo salvo com extensão “.lp” que estava sendo executado quando a memória do problema 38 esgotou tinha tamanho maior que 1Gb. Isso ocorre porque o número de variáveis sendo usados nessa abordagem é muito maior e a permutação dos carros faz os modelos do espaço complementar crescerem muito e muito rápido.

Apesar do caráter heurístico desses testes, foi possível provar 2 valores ótimos para as instâncias 32 e 14, visto que cada RS foi explorada completamente e também o espaço complementar final. De modo geral, a quantidade de espaços explorados não foi tão grande quanto no IRSS exato. Sendo que a quantidade máxima de RS explorados em alguma instância foi 10 na *pgdbcm11*, que ainda teve o espaço complementar avaliado. Se comparados os desempenhos com os vindos da TABELA 4.4 essa abordagem conseguiu atingir 25 resultados melhores ou iguais dos 39 possíveis. A comparação entre todas as abordagens está detalhada no próximo capítulo.

GRÁFICO 5.2 - QUANTIDADE DE MELHORES SOLUÇÕES ALCANÇADAS POR ITERAÇÃO COM O IRSS HEURÍSTICO



FONTE: o autor (2018)

O GRÁFICO 5.2 traz um resumo mostrando em qual iteração RS foi possível atingir a melhor solução. Como pode ser observado, em 13 ocasiões a melhor solução obtida ocorreu na primeira iteração, em outras 9 aconteceu na terceira iteração, na segunda e quarta iterações o melhor resultado encontrado apareceu 7 vezes e na quinta iteração o melhor resultado foi encontrado 3 vezes. Em nenhum

teste a melhor solução encontrada apareceu após a avaliação do quinto espaço restrito. Isso mostra o bom desempenho do algoritmo já nas primeiras iterações e que a combinação com o ACO faz com que não seja necessário a exploração de muitos espaços restritos para conseguir um bom resultado.

Como a memória esgotou nos testes das instâncias *pgdbcm22* e *pgdbcm38*, testaram-se as instâncias em um *workstation* com processador Intel Xeon com processamento de 3.6 Ghz, 64Gb de RAM e sistema operacional 64 bits. A instância 22 permaneceu com o mesmo resultado e a 38 foi melhorada para 426, os resultados estão à disposição na TABELA 5.4.

TABELA 5.4 - RESPOSTAS DO IRSS HEURÍSTICO PARA AS INSTÂNCIAS *pgdbcm22* E *pgdbcm38* EM UM WORKSATON

Instâncias	IRSS - Espaços Restritos					IRSS - Complementar			IRSS Geral		
	Tempo	Solução	Iteração	Critério de Parada	Todos ótimos	Tempo	Lower Bound	Solução	Ótimo	Solução	Tempo Acum.
<i>pgdbcm22</i>	74953	426	1	1	Não	11446	-	81480	Não	426	86572
<i>pgdbcm38</i>	76312	426	2	1	Não	10088	-	30512	Não	426	86572

FONTE: o autor (2018)

A comparação entre todos os resultados e a eficiência do IRSS heurístico será melhor explorada no capítulo a seguir, onde todas as outras técnicas aqui aplicadas serão colocadas lado a lado.

6 RESULTADOS GERAIS

Essa seção tem por objetivo condensar todos os resultados alcançados com cinco das abordagens apresentadas ao longo desse trabalho. Quatro delas, usando modelagem matemática com *solver*, tiveram um tempo limite de 24 horas de resolução para, de forma justa, identificar qual das estratégias aparentasse ser a mais promissora. Os resultados aqui apresentados são referentes ao novo modelo proposto nessa tese, visto que o antigo modelo exigiria uma diferente abordagem pelo algoritmo ACO e a comparação não seria justa. As cinco alternativas que serão comparadas serão:

1. *Solver* Gurobi limitado em 24 horas;
2. O algoritmo ACO limitado em 1 hora ou 1 iteração completa;
3. *Solver* Gurobi limitado em 23 horas usando ACO limitado em 1 hora;
4. IRSS Exato limitado em 24 horas, com o ACO limitado em 10 minutos e cada RS tendo 10 minutos para ser completamente explorado;
5. IRSS Heurístico com ACO limitado em 10 minutos com cada RS podendo ser explorado por no máximo 2 horas sem melhorias ou 6 horas totais.

As tabelas, TABELA 6.1 e TABELA 6.2 mostram todos os resultados das cinco abordagens para as instâncias de *pgdbcm1* a *pgdbcm20* e de *pgdbcm21* a *pgdbcm39* respectivamente. As colunas estão organizadas de modo a mostrar os valores de *lower bounds* e da solução sempre que possível, ao final é mostrado os melhores resultados alcançados.

Os *solvers* comerciais, como o Gurobi, têm performances muito boas e chegam a misturar heurísticas em seus métodos, de modo que o desempenho é difícil de ser superado. Então, alinhar conhecimento específico de cada problema com a resolução exata do problema é uma alternativa muito boa. Isto foi feito nas últimas três alternativas de resolução que traziam contribuições da ACO. Vale lembrar que a colônia de formigas é um algoritmo heurístico que gera soluções de maneira estocástica e, assim, cada vez que é executado pode trazer respostas diferentes. De fato, no PCARP-CM, o algoritmo traz respostas diferentes na maioria das vezes em virtude do sorteio do ponto inicial.

TABELA 6.1 - RESULTADOS GERAIS DO PCARP-CM INSTÂNCIAS *pgdbcm1* A *pgdbcm20*

Instâncias	<i>Solver</i> (1)		ACO (2)	ACO + <i>Solver</i> (3)		IRSS Exato (4)		IRSS Heurístico (5)		Geral				
	Bound	Solução		Bound	Solução	Bound	Solução	Bound	Solução	Melhor bound usando	Lower bound	Melhor solução usando	Solução	GAP
<i>pgdbcm1</i>	451	451	451	451	451	451	451	-	451	1; 3; 4;	451	1; 2; 3; 4; 5;	451 *	0,0%
<i>pgdbcm2</i>	518	614	3594	519	582	511	604	-	576	3;	519	5;	576	9,9%
<i>pgdbcm3</i>	315	340	344	331	340	324	340	318	340	3;	331	1; 3; 4; 5;	340	2,6%
<i>pgdbcm4</i>	449	2519	2519	453	2546	447	1560	-	1560	3;	453	4; 5;	1560	71,0%
<i>pgdbcm5</i>	659	6737	9764	659	5742	659	2768	-	7757	3;	659	4;	2768	76,2%
<i>pgdbcm6</i>	418	456	456	418	456	418	456	418	456	4;	418	1; 2; 3; 4; 5;	456	8,3%
<i>pgdbcm7</i>	494	508	576	496	508	494	508	-	516	3;	496	1; 3; 4;	508	2,4%
<i>pgdbcm8</i>	325	8395	22364	325	5385	326	6363	-	26359	4;	326	3;	5385	93,9%
<i>pgdbcm9</i>	439	44425	31446	439	30443	439	20459	-	33472	4;	439	4;	20459	97,9%
<i>pgdbcm10</i>	684	766	3801	684	733	684	751	-	5744	4;	684	3;	733	6,7%
<i>pgdbcm11</i>	416	442	442	413	442	413	442	-	442	1;	416	1; 2; 3; 4; 5;	442	5,9%
<i>pgdbcm12</i>	484	536	536	481	536	471	527	-	527	1;	484	4; 5;	527	8,2%
<i>pgdbcm13</i>	620	620	620	620	620	620	620	-	620	1; 3; 4;	620	1; 2; 3; 4; 5;	620 *	0,0%
<i>pgdbcm14</i>	145	145	149	145	145	145	145	145	145	1; 3; 4; 5;	145	1; 3; 4; 5;	145 *	0,0%
<i>pgdbcm15</i>	137	140	1152	137	140	137	140	-	140	4;	137	1; 3; 4; 5;	140	2,1%
<i>pgdbcm16</i>	176	182	204	176	182	176	182	-	182	3; 4;	176	1; 3; 4; 5;	182	3,3%
<i>pgdbcm17</i>	161	168	183	161	168	161	168	-	168	1; 3; 4;	161	1; 3; 4; 5;	168	4,2%
<i>pgdbcm18</i>	369	379	2447	369	377	369	377	-	369	1; 3; 4;	369	5;	369 *	0,0%
<i>pgdbcm19</i>	125	125	131	125	125	122	125	125	125	1; 3; 5;	125	1; 3; 4; 5;	125 *	0,0%
<i>pgdbcm20</i>	240	2262	4278	247	258	241	261	-	2279	3;	247	3;	258	4,3%

FONTE: o autor (2018)

TABELA 6.2 - RESULTADOS GERAIS DO PCARP-CM INSTÂNCIAS *pgdbcm21* A *pgdbcm39*

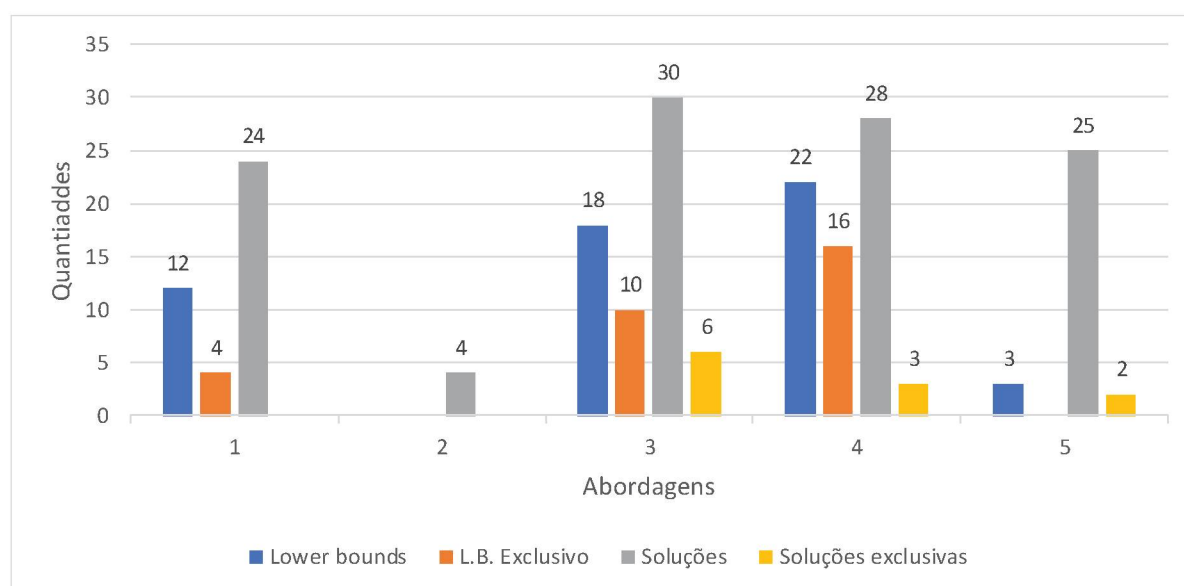
Instâncias	<u>Solver (1)</u>		ACO (2)	<u>ACO + Solver (3)</u>		<u>IRSS Exato (4)</u>		<u>IRSS Heurístico (5)</u>		<u>Geral</u>				
	Bound	Solução		Bound	Solução	Bound	Solução	Bound	Solução	Melhor bound usando	bound	Melhor solução usando	Solução	GAP
<i>pgdbcm21</i>	210	212	222	206	212	204	212	-	212	1;	210	1; 3; 4; 5;	212	0,9%
<i>pgdbcm22</i>	423	435	1492	423	426	423	427	-	426	4;	423	3; 5;	426	0,7%
<i>pgdbcm23</i>	329	336	357	329	336	329	336	-	336	4;	329	1; 3; 4; 5;	336	2,1%
<i>pgdbcm24</i>	517	582	620	520	576	511	574	-	616	3;	520	4;	574	9,4%
<i>pgdbcm25</i>	323	340	370	323	340	315	340	315	340	1; 3;	323	1; 3; 4; 5;	340	5,0%
<i>pgdbcm26</i>	447	508	573	448	525	447	508	-	519	3;	448	1; 4;	508	11,8%
<i>pgdbcm27</i>	639	676	4789	639	680	639	676	-	676	3;	639	1; 4; 5;	676	5,5%
<i>pgdbcm28</i>	402	444	447	418	444	411	444	-	444	3;	418	1; 3; 4; 5;	444	5,9%
<i>pgdbcm29</i>	305	339	3415	305	339	305	345	-	349	4;	305	1; 3;	339	10,0%
<i>pgdbcm30</i>	375	21449	25424	375	8442	378	17439	-	21455	4;	378	3;	8442	95,5%
<i>pgdbcm31</i>	684	748	4906	684	737	684	743	-	755	4;	684	3;	737	7,2%
<i>pgdbcm32</i>	523	523	536	491	523	479	523	523	523	1; 5;	523	1; 3; 4; 5;	523	* 0,0%
<i>pgdbcm33</i>	153	157	158	153	157	154	157	-	157	4;	154	1; 3; 4; 5;	157	1,9%
<i>pgdbcm34</i>	120	120	131	117	120	119	120	-	120	1;	120	1; 3; 4; 5;	120	* 0,0%
<i>pgdbcm35</i>	167	170	209	167	170	167	170	-	170	4;	167	1; 3; 4; 5;	170	1,8%
<i>pgdbcm36</i>	369	381	2479	369	371	369	379	-	371	4;	369	3; 5;	371	0,5%
<i>pgdbcm37</i>	240	264	4284	244	261	246	264	-	264	4;	246	3;	261	5,7%
<i>pgdbcm38</i>	423	426	525	423	426	423	426	-	428	4;	423	1; 3; 4;	426	0,7%
<i>pgdbcm39</i>	329	336	380	329	336	329	336	-	339	4;	329	1; 3; 4;	336	2,1%
Média	374	2530	3507	375	1682	373	1581		2839		376		1323	

FONTE: o autor (2018)

Todos os testes foram feitos em notebooks Intel Core i7-5500U de 2.4 GHz com 8Gb de RAM e sistema operacional 64 bits, a linguagem de programação utilizada tanto para utilização do Gurobi, quanto para programação do ACO e a alternância entre ambos é a VisualBasic.net. Foram utilizadas aproximadamente 3500 horas de processamento para todos os testes presentes nas duas tabelas, os tempos foram omitidos, mas podem ser encontrados nas respectivas seções. Todo esforço permitiu mostrar que os resultados apresentados unicamente pelo *solver* (1) foram atingidos ou superados em todas as instâncias. Algumas instâncias que parecem mais complicadas de serem resolvidas, visto valores da função objetivo com suas penalidades por atraso, por exemplo, *pgdbcm8* e *pgdbcm9*, foram muito melhoradas.

Se avaliados os valores médios alcançados, a técnica que trouxe o resultado mais baixo foi o IRSS Exato (4), enquanto que o *solver* utilizando a solução inicial vinda do ACO conseguiu aumentar os valores médios dos *lower bounds*. O pior desempenho ficou para o ACO limitado em 1 hora ou ao menos 1 iteração, mesmo assim o resultado é interessante e pode ser uma alternativa ao alto custo de um *solver* comercial. Tanto o desempenho do *solver* (1) quanto o da IRSS heurística (2) são próximos em questão de valores médios.

GRÁFICO 6.1 - ANÁLISE DAS CINCO ABORDAGENS EM NÚMEROS



FONTE: o autor (2018)

O GRÁFICO 6.1 traz algumas informações importantes dos dados das tabelas, ela contém os números alcançados com cada abordagem. A primeira barra, azul, mostra quantos melhores *lower bounds* foram alcançados com a técnica e a

segunda barra, em laranja, quantos foram exclusivos, isto é, apenas a técnica conseguiu. A quantidade de melhores soluções alcançadas está em cinza e a quantidade exclusiva da técnica em amarelo.

O ACO sozinho (2) foi capaz de atingir os melhores resultados conhecidos em apenas 4 das 39 instâncias, porém sua contribuição foi valiosa nas hibridizações. O *solver* sozinho (1) trouxe 4 *lower bounds* exclusivos e nenhuma melhor solução exclusiva, estando à frente apenas do ACO quando tratando do alcance das melhores soluções, encontrando 24 delas.

Dentre as hibridizações, a que menos se destaca é o IRSS Heurístico (5) que conseguiu 25 melhores soluções, sendo duas exclusivas. Um destaque deve ser dado para a instância *pgdbcm18* que atingiu o *lower bound* proposto em outras técnicas, levando a um resultado ótimo de 369 que ainda não havia sido provado. Poucos *lower bounds* foram alcançados, dos 6 encontrados apenas metade estiveram entre os maiores.

Destaque deve ser dado à utilização do algoritmo ACO gerando soluções iniciais para o *solver* e ao uso do IRSS Exato. Ambas abordagens exatas e que conseguiram baixar *lower bounds* alcançaram um alto número de melhores soluções, 30 e 28 respectivamente. O IRSS Exato conseguiu reduzir 16 *lower bounds* e 3 melhores soluções exclusivas. Já o *solver* com ACO foi responsável por novos 10 *lower bounds* exclusivos e 6 melhores soluções.

7 CONSIDERAÇÕES FINAIS

O Problema de Roteamento Periódico e Capacitado em Arcos foi o foco dessa tese. É uma classe de problemas recente, com menos de 20 anos de história, e rica em aplicabilidade. Aqui foram abordados todos os trabalhos que dão destaque ao PCARP e que se encontram nas bases de dados *Web-of-science* e *Scopus*. Isso permitiu a primeira contribuição científica, a classificação taxonômica de cada artigo levando em conta aspectos físicos, foco teórico ou aplicado e abordagem de resolução. As características físicas envolvem períodos, horizonte de tempo, tipo de serviço, tipo de roteirização, dimensionamento da frota, como funcionam as frequências, como é o rigor da demanda e a natureza dos grafos.

O estudo bibliométrico mostra que o assunto é recorrente na literatura e diante de tantas características, como as aqui listadas, é notável o campo fértil para pesquisa na área. Esta pesquisa aprofundou-se no PCARP com Movimentos Contínuos, com uma aplicação já estudada em Batista (2014), mas que não havia sido classificada dessa forma. O movimento contínuo é uma característica comumente encontrada em problemas de *pick-up and delivery* e que se encaixa muito bem na inspeção e manutenção de ferrovias.

Uma alternativa a modelagem matemática feita em 2014 expande o espaço solução do PCARP-CM, possibilitando alcançar uma nova gama de respostas. A característica cíclica do PCARP-CM é mais explorada por esse novo modelo, em compensação a complexidade aumenta permitindo a troca de pontos iniciais entre os carros ao final do horizonte de tempo. A aplicação é importante, pois visa garantir que trechos de uma ferrovia sejam avaliados dentro de suas periodicidades previamente definidas, ao passo que também tenta reduzir os custos de deslocamento. A solução encontrada acaba formando uma política permanente, sendo que alterações na malha ferroviária conduziriam a um novo problema. Atender os arcos dentro dos limites estabelecidos ajuda a evitar grandes prejuízos e acidentes que podem decorrer, por exemplo, de um descarrilamento.

Os atuais *solvers* comerciais têm demonstrado bom desempenho para resolver uma série de problemas e o mesmo acontece no PCARP-CM. Aproveitou-se dessa característica para a criação de métodos híbridos na busca por melhores soluções dado um intervalo de tempo de 24 horas. A hibridização envolveu ao

menos dois algoritmos: o Algoritmo de Otimização de Colônia de Formigas (ACO) e o método exato usado pelo *solver* Gurobi.

O ACO demonstrou uma boa e rápida performance para resolver o PCARP-CM. A proposta foi inovadora mesclando ideias vindas da Busca Tabu Granular com Busca Local. Sem essa mescla o desempenho do algoritmo não seria tão efetivo. Movimentos não tão interessantes foram desconsiderados na geração de soluções. Após geradas, cada solução foi melhorada com a busca em vizinhanças pré-definidas. O algoritmo ainda demonstrou pontos fortes em curto prazo ao atingir resultados melhores que o *solver* em problemas críticos nos quais a taxa de ocupação dos veículos é alta. Essa característica é algo a ser aproveitado para hibridizações.

A simples aplicação do ACO como gerador de soluções iniciais para o *solver* trouxe resultados que até então não haviam sido alcançados. A fim de superar esses resultados, outra hibridização mais elaborada foi feita: o espaço solução foi fragmentado e explorado iterativamente pelo algoritmo de Busca Iterativa em Espaço Restrito (IRSS). Cada espaço a ser explorado é um pedaço do espaço completo que foi restrito, entretanto a definição de cada espaço não é uma tarefa trivial.

Para a geração de espaços restritos ser eficiente é desejável um algoritmo que explore bem o espaço solução e de maneira diversificada. O ACO traz todas as características para o bom desempenho do IRSS, além de uma riqueza de informações com a matriz de feromônios. A matriz de feromônios traz a história da exploração feita por cada formiga que despejou feromônios em seus caminhos. Esses dados permitiram a criação dos espaços restritos usados no algoritmo IRSS.

O IRSS é um algoritmo recente que provou seu valor para o PCARP-CM sendo aplicado de duas formas diferentes, exata e heurística. A abordagem exata é complexa, pois exige que valores ótimos sejam encontrados dentro de cada espaço restrito. A estratégia encontrada para fazer com que isso ocorra foi a diminuição gradual do espaço restrito baseado na fixação de uma porcentagem de variáveis vindas do ACO. Estratégia essa que permitiu com que muitos espaços fossem explorados trazendo ganhos relevantes para o problema, como valores mais altos de *lower bounds* e funções objetivo mais baixas. A abordagem heurística é mais simples, porém exigiu bastante da memória dos notebooks usados em virtude da grande quantidade de variáveis. Vale ressaltar que um passo importante da

aplicação da IRSS é a identificação de soluções que são simétricas e que devem ser evitadas. Elas são muitas e acarretam em uma mesma sequência de atendimentos alternando carros, sentido e dias de atendimento.

Ainda existe muito a ser explorado no que diz respeito ao PCARP e ao PCARP-CM. O PCARP tem uma série de combinações de características ainda não estudadas e os trabalhos recentes têm mostrado evoluções significativas no que diz respeito aos períodos. Quanto ao PCARP-CM, a definição do tamanho do horizonte de tempo é algo que é pré-definido, porém pode ser otimizado visto que seu tamanho tem grande influência no resultado das rotas. Novos modelos podem ser estruturados considerando outras variáveis de decisão e outros objetivos como minimização de número de veículos ou atrasos. Como é uma classe de problemas recentes, existe abertura para aplicação de variadas técnicas de Pesquisa Operacional na tentativa de alcançar resultados melhores. Além disso, muitas variantes podem vir a surgir com novas aplicações.

Espera-se que este trabalho ajude no desenvolvimento de novas pesquisas. A classificação taxonômica pode auxiliar nessa tarefa e ainda ser incrementada conforme os problemas forem evoluindo. As técnicas aqui aplicadas podem ser usadas em muitos outros problemas bastando uma adaptação das ideias utilizadas. A Pesquisa Operacional tem contribuído muito na resolução de problemas logísticos como o PCARP-CM e uma série de outros problemas, aqui um passo a mais foi dado inovando em técnica de resolução e aplicação.

REFERÊNCIAS

AMAYA, A.; LANGEVIN, A.; TRÉPANIÉ, M. The capacitated arc routing problem with refill points. **Operations Research Letters**, v. 35, n. 1, p. 45–53, 2007. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S016763770600006X>>. Acesso em: 28/8/2013.

ANDERSSON, H.; HOFF, A.; CHRISTIANSEN, M.; HASLE, G.; LØKKETANGEN, A. Industrial aspects and literature survey: Combined inventory management and routing. **Computers and Operations Research Research**, v. 37, n. 9, p. 1515–1536, 2010. Elsevier. Disponível em: <<http://dx.doi.org/10.1016/j.cor.2009.11.009>>. .

ASSAD, A. A.; GOLDEN, B. L. Arc Routing Methods and Applications. **Handbooks in Operations Research and Management Science, volume 8, Network Routing**. v. 8, p.375–483, 1995.

BARTOLINI, E.; CORDEAU, J.; LAPORTE, G. An Exact Algorithm for the Capacitated Arc Routing Problem with Deadheading Demand An Exact Algorithm for the Capacitated Arc Routing Problem with Deadheading Demand. **Operations Research**, v. 61, n. 2, p. 315–327, 2013.

BATISTA, G. V. **PROPOSTA DE UM MODELO MATEMÁTICO PARA O PROBLEMA DE ROTEAMENTO EM ARCOS CAPACITADO E PERIÓDICO**, 2014. Universidade Federal do Paraná.

BATISTA, G. V.; SCARPIN, C. T. Comparison between two mathematical models for problems in monitoring and inspection of arcs. **International Refereed Journal of Engineering and Science**, v. 3, n. 6, p. 19–24, 2014.

BATISTA, G. V.; SCARPIN, C. T. PROBLEMA DE ROTEAMENTO EM ARCOS CAPACITADO E PERIÓDICO APLICADO A UM CONTEXTO REAL. **Revista Produção Online**, v. 15, n. 3, p. 1080–1098, 2015.

BELENGUER, J.-M.; BENAVENT, E.; LABADI, N.; PRINS, C.; REGHIOUI, M. Split-Delivery Capacitated Arc-Routing Problem: Lower Bound and Metaheuristic. **Transportation Science**, v. 44, n. 2, p. 206–220, 2010. Disponível em: <<http://pubsonline.informs.org/doi/abs/10.1287/trsc.1090.0305>>. Acesso em: 13/3/2014.

BELENGUER, J.; BENAVENT, E. A cutting plane algorithm for the capacitated arc routing problem. **Computers & Operations Research**, v. 30, p. 705–728, 2003. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054802000461>>. Acesso em: 20/1/2014.

BENAVENT, E.; CAMPOS, V.; CORBERÁN, A.; MOTA, E. The capacitated chinese postman problem: lowerbounds. **Networks**, v. 22, n. 7, p. 669–690, 1992.

BEULLENS, P.; MUYLDERMANS, L.; CATTRYSSE, D.; OUDHEUSDEN, D. VAN. A guided local search heuristic for the capacitated arc routing problem. **European Journal of Operational Research**, v. 147, p. 629–643, 2003.

BRANDÃO, J.; EGGLESE, R. A deterministic tabu search algorithm for the capacitated arc routing problem. **Computers & Operations Research**, v. 35, n. 4, p. 1112–1126, 2008.

CHEN, Y.; HAO, J. K. Two phased hybrid local search for the periodic capacitated arc routing problem. **European Journal of Operational Research**, v. 264, n. 1, p. 55–65, 2018. Elsevier B.V.

CHRISTOFIDES, N.; BEASLEY, J. E. The Period Routing Problem. **Networks**, v. 14, p. 237–256, 1984.

CHU, F.; LABADI, N.; PRINS, C. Periodic arc routing problems: Linear programming model and heuristics. 9th Intern. Multiconf. on Advanced Computers Systems. **Anais...** . p.10, 2002. Miedzyzdroje (Pologne). Disponível em: <<http://cat.inist.fr/?aModele=afficheN&cpsidt=14404443>>. Acesso em: 19/11/2016.

CHU, F.; LABADI, N.; PRINS, C. Lower bounds for the Periodic Capacitated Arc Routing Problem. Odysseus 2003 (2nd Int. Workshop on Freight Transportation and Logistics). **Anais...** . p.27–30, 2003. Palerme (Italie).

CHU, F.; LABADI, N.; PRINS, C. THE PERIODIC CAPACITATED ARC ROUTING PROBLEM LINEAR PROGRAMMING MODEL, METAHEURISTIC AND LOWER BOUNDS. **Journal of Systems Science and Systems Engineering**, v. 13, n. 4, p. 423–435, 2004a.

CHU, F.; LABADI, N.; PRINS, C. A scatter search for the periodic capacitated arc routing problem. Project Management and Scheduling (PMS). **Anais...** . p.415–420, 2004b. Nancy, France.

CHU, F.; LABADI, N.; PRINS, C. The Periodic Capacitated Arc Routing Problem: linear programming model, metaheuristic and lower bound. ICSSSM 2004 (Int. Conf. on Service Systems and Service Management). **Anais...** . p.784–789, 2004c. Beijing, China: World Publishing Co.

CHU, F.; LABADI, N.; PRINS, C. Heuristics for the periodic capacitated arc routing problem. **Journal of Intelligent Manufacturing**, v. 16, n. 2, p. 243–251, 2005. Disponível em: <<http://link.springer.com/article/10.1007/s10845-004-5892-8>>. .

CHU, F.; LABADI, N.; PRINS, C. A Scatter Search for the periodic capacitated arc routing problem. **European Journal of Operational Research**, v. 169, p. 586–605, 2006.

COLORNI, A.; DORIGO, M.; MANIEZZO, V. Distributed Optimization by Ant Colonies. **Ecal91 - European Conference on Artificial Life**, , n. January 1991, p. 132–142, 1991.

CORBERÁN, A.; MOTA, E.; SANCHIS, J. M. A comparison of two different formulations for arc routing problems on mixed graphs. **Computers & Operations Research**, v. 33, p. 3384–3402, 2006.

CORBERÁN, A.; PRINS, C. Recent Results on Arc Routing Problems : An Annotated Bibliography. **Networks**, p. 50–69, 2010.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos: teoria e prática**. 2a edição ed. Rio de Janeiro, 2002.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant System: Optimization by a Colony of Cooperating Agents. , v. 26, n. 1, p. 1–13, 1996. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=484436&tag=1>. .

DORIGO, M.; STÜTZLE, T. Ant Colony Optimization: Overview and Recent Advances. In: M. Gendreau; J.-Y. Potvin (Orgs.); **Handbook of Metaheuristics**. Second Edi ed., v. 146, p.227–263, 2010. Springer.

DROR, M. **Arc routing : theory , solutions , and applications**. Boston, Mass. [u.a.]: Kluwer Academic Publishers, 2000.

EGLESE, R. W. Routeing winter gritting vehicles. **Discrete Applied Mathematics**, v. 48, n. 3, p. 231–244, 1994. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/0166218X92000035>>. .

EISELT, H. A.; GENDREAU, M.; LAPORTE, G. Arc Routing Problems , Part II : The Rural Postman Problem. **Operations Research**, v. 43, n. 3, p. 399–414, 1995a. Disponível em: <<http://or.journal.informs.org/content/43/3/399.full.pdf>>. .

EISELT, H. A.; GENDREAU, M.; LAPORTE, G. Arc Routing Problems, Part I: The Chinese Postman Problem. **Operations Research**, v. 43, n. 3, p. 231–242, 1995b. Disponível em: <<http://or.journal.informs.org/cgi/doi/10.1287/opre.43.2.231>>. .

EKSIOGLU, B.; VURAL, A. V.; REISMAN, A. The vehicle routing problem: A taxonomic review. **Computers & Industrial Engineering**, v. 57, n. 4, p. 1472–1483, 2009. Elsevier Ltd. Disponível em: <<http://dx.doi.org/10.1016/j.cie.2009.05.009>>. .

FERNÁNDEZ, E.; MEZA, O.; GARFINKEL, R.; ORTEGA, M. ON THE UNDIRECTED RURAL POSTMAN PROBLEM : TIGHT BOUNDS BASED ON A NEW FORMULATION. **Operations Research**, v. 51, n. 2, p. 281–291, 2003.

FISCHETTI, M.; LODI, A. Local branching. **Mathematical Programming**, v. 98, n. 1–3, p. 23–47, 2003.

FRANCIS, P.; SMILOWITZ, K. Modeling techniques for periodic vehicle routing problems. **Transportation Research Part B**, v. 40, p. 872–884, 2006.

FUNG, R. Y. K.; LIU, R.; JIANG, Z. A memetic algorithm for the open capacitated arc routing problem. **Transportation Research Part E: Logistics and Transportation Review**, v. 50, p. 53–67, 2013. Elsevier Ltd. Disponível em:

<<http://linkinghub.elsevier.com/retrieve/pii/S1366554512000932>>. Acesso em: 28/8/2013.

GEOFFRION, A. M. Elements of Large-Scale Mathematical Programming: Part I: Concepts. **Management Science**, v. 16, n. 11, p. 652–675, 1970a. Disponível em: <<http://www.jstor.org.eproxy1.lib.hku.hk/stable/2628901>>. .

GEOFFRION, A. M. Elements of Large Scale Mathematical Programming Part II : Synthesis of Algorithms and Bibliography. **Management Science**, v. 16, n. 11, p. 676–691, 1970b.

GHIANI, G.; IMPROTA, G. An algorithm for the hierarchical Chinese postman problem. **Operations Research Letters**, v. 26, n. 1, p. 27–32, 2000. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0167637799000462>>. .

GHIANI, G.; MUSMANNO, R.; PALETTA, G.; TRIKI, C. A heuristic for the periodic rural postman problem. **Computers & Operations Research**, v. 32, n. 2, p. 219–228, 2005. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0305054803002132>>. Acesso em: 28/8/2013.

GOLDEN, B. L.; DEARMON, J. S.; BAKER, E. K. Computational experiments with algorithms for a class of routing problems. **Computers & Operations Research**, v. 10, n. 1, p. 47–59, 1983. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0305054883900266>>. Acesso em: 13/3/2014.

GOLDEN, B. L.; WONG, R. T. Capacitated Arc Routing Problem. **Networks**, v. 11, p. 305–311, 1981.

GUIMARÃES, T. A. **ABORDAGEM DETERMINÍSTICA E ESTOCÁSTICA NA FORMULAÇÃO DE POLÍTICAS DE DISTRIBUIÇÃO POR LOTE ECONÔMICO DE ENTREGA, EM PROBLEMAS DE ROTEIRIZAÇÃO COM ESTOQUE GERENCIADO PELO FORNECEDOR E SISTEMA LOGÍSTICO EM TRÊS NÍVEIS**, 2015. Universidade Federal do Paraná.

HERTZ, A.; LAPORTE, G.; MITTAZ, M. A Tabu Search heuristic for the Capacitated Arc Routing Problem. **Operations Research**, v. 48, n. 1, p. 129–135, 2000.

HSIAO-FAN, W.; YU-PIN, W. Time-Constrained Chinese Postman Problems. **Computer and Mathematics with Applications** 44, v. 44, n. 02, p. 375–387, 2002.

HUANG, S.; LIN, T. Using Ant Colony Optimization to solve Periodic Arc Routing Problem with Refill Points. , , n. January 2015, p. 37–41, 2014.

JURAN, J. M.; GODFREY, A. B. **Juran's Quality Handbook**. 5th ed. New York: McGraw-Hill, 1998.

KANSOU, A.; YASSINE, A. Ant Colony System for the Periodic Capacitated Arc Routing Problem. International Network Optimization Conference. **Anais...** . p.1–7, 2009.

KONOWALENKO, F. **PROBLEMA DO CARTEIRO CHINÊS NÃO-ORIENTADO E MISTO PARA A OTIMIZAÇÃO DE ROTAS NA CIDADE DE IRATI / PR**, 2012. Universidade Federal do Paraná.

LACOMME, P.; PRINS, C.; RAMDANE-CHÉRIF, W. General arc routing problems solved by a cutting plane algorithm and a genetic algorithm. In: IFAC (Org.); 15th Triennial World Congress, Barcelona, Spain. **Anais...** , 2002a. Barcelona, Spain.

LACOMME, P.; PRINS, C.; RAMDANE-CHÉRIF, W. Evolutionary Algorithms for Multiperiod Arc Routing Problems. In: IPMU 2002 (Org.); 9th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based systems. **Anais...** . p.1–8, 2002b. Annecy, France: ESIA-University of Savoie.

LACOMME, P.; PRINS, C.; RAMDANE-CHÉRIF, W. Planning Problems in Arc Routing. In: F. U.-E. de Valencia (Org.); Actes de PMS 2002 (EURO Working Group on Project Management and Scheduling). **Anais...** . p.232–235, 2002c. Valencia, Espanha.

LACOMME, P.; PRINS, C.; RAMDANE-CHÉRIF, W. Evolutionary algorithms for periodic arc routing problems. **European Journal of Operational Research**, v. 165, p. 535–553, 2005.

LAPORTE, G.; MUSMANNO, R.; VOCATURO, F. An Adaptive Large Neighbourhood Search Heuristic for the Capacitated Arc-Routing Problem with Stochastic Demands. **Transportation Science**, v. 44, n. 1, p. 125–135, 2010. Disponível em: <<http://transci.journal.informs.org/cgi/doi/10.1287/trsc.1090.0290>>. Acesso em: 28/8/2013.

LONGO, H.; DE ARAGÃO, M. P.; UCHOA, E. Solving capacitated arc routing problems using a transformation to the CVRP. **Computers & Operations Research**, v. 33, n. 6, p. 1823–1837, 2006. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0305054804003223>>. Acesso em: 28/8/2013.

LOPES, R. B.; FERREIRA, C.; SOUSA, B. A taxonomical analysis , current methods and objectives on location-routing problems. **International Transactions in Operational Research**, v. 00, p. 1–29, 2013.

LOPES, T. C.; SIKORA, C. G. S.; MAGATÃO, L. Buffer and Cyclical Product Sequence Aware Assembly Line Balancing Problem: Model and Steady-State Balancing Case Study. Annals of the XLVIII SBPO. **Anais...** . p.3458–3469, 2016.

MALANDRAKI, C.; DASKIN, M. S. Theory and Methodology The maximum benefit Chinese postman problem and the maximum benefit traveling salesman problem. **European Journal of Operational Research**, v. 65, n. 1093, p. 218–234, 1993.

MARZOLF, F.; TRÉPANIÉ, M.; LANGEVIN, A. Road network monitoring: algorithms and a case study. **Computers & Operations Research**, v. 33, n. 12, p. 3494–3507, 2006. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0305054805000717>. Acesso em: 20/1/2014.

MEI, Y.; MEMBER, S.; TANG, K.; YAO, X. A Memetic Algorithm for Periodic Capacitated Arc Routing Problem. , v. 41, n. 6, p. 1654–1667, 2011.

MONROY, I. M.; AMAYA, C. A.; LANGEVIN, A. The periodic capacitated arc routing problem with irregular services. **Discrete Applied Mathematics**, v. 161, p. 691–701, 2013.

PEARL, J. **Heuristics: Intelligent Search Strategies for Computer Problem Solving**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.

PÉCORA JR, J. E. **Iterative Restricted Space Search: a solving approach based on hybridization**, 2008. Université Laval.

PÉCORA JR, J. E.; RUIZ, A.; SORIANO, P. Minimization of the Wood Density Variation in Pulp and Paper Production. **INFOR: Information Systems and Operational Research**, v. 45, n. 4, p. 187–196, 2007.

RIQUELME-RODRÍGUEZ, J.-P.; GAMACHE, M.; LANGEVIN, A. Adaptive Large Neighborhood Search for the Periodic Capacitated Arc Routing Problem with Inventory Constraints. **Networks**, v. 47, n. 1, p. 26–36, 2014a.

RIQUELME-RODRÍGUEZ, J.-P.; GAMACHE, M.; LANGEVIN, A. Periodic capacitated arc-routing problem with inventory constraints. **Journal of the Operational Research Society**, v. 65, n. 12, p. 1840–1852, 2014b. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84925938293&partnerID=tZOtx3y1>. .

RIQUELME-RODRÍGUEZ, J.-P.; GAMACHE, M.; LANGEVIN, A. Location arc routing problem with inventory constraints. **Computers & Operations Research**, v. 76, p. 84–94, 2016.

ROSA, B. DE; IMPROTA, G.; GHIANI, G.; MUSMANNO, R. The Arc Routing and Scheduling Problem with Transshipment. **Transportation Science**, v. 36, n. 3, p. 301–313, 2002.

SAVELSBERGH, M.; SONG, J. Inventory routing with continuous moves. , v. 34, p. 1744–1763, 2007.

SAVELSBERGH, M.; SONG, J. An optimization algorithm for the inventory routing problem with continuous moves. , v. 35, p. 2266–2282, 2008.

SCHNEEWEISS, C. **Distributed Decision Making**. 2 edition ed. New York, 2003.

TAGMOUTI, M.; GENDREAU, M.; POTVIN, J.-Y. Arc routing problems with time-dependent service costs. **European Journal of Operational Research**, v. 181, n. 1, p. 30–39, 2007. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0377221706004644>>. Acesso em: 28/8/2013.

TAUBE, J. DE M. **O problema do Carteiro Chinês**, 1992. UNICAMP.

TIRKOLAEI, E. B.; MAHDAVI, I.; MEHDI SEYYED ESFAHANI, M. A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time. **Waste Management**, v. 76, p. 138–146, 2018. Elsevier Ltd. Disponível em: <<https://doi.org/10.1016/j.wasman.2018.03.015>>. .

TORTELLY JR., A. C.; OCHI, L. S. Uma metaheurística híbrida GRASP+Tabu para o problema de roteamento periódico em uma frota de veículos. XXXV Simpósio Brasileiro de Pesquisa Operacional. **Anais...** . p.1–11, 2003. Natal-RN.

TOTH, P.; VIGO, D. The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. **INFORMS Journal on Computing**, v. 15, n. 4, p. 333–346, 2003. Disponível em: <<http://pubsonline.informs.org/doi/abs/10.1287/ijoc.15.4.333.24890>>.

USBERTI, F. L. Métodos Heurísticos e Exatos para o Problema de Roteamento em Arcos Capacitado e Aberto. , 2012.

USBERTI, F. L.; FRANÇA, P. M.; FRANÇA, A. L. M. The open capacitated arc routing problem. **Computers & Operations Research**, v. 38, n. 11, p. 1543–1555, 2011. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0305054811000244>>. Acesso em: 28/8/2013.

VIDAL, T.; CRAINIC, T. G.; GENDREAU, M.; LAHRICHI, N.; REI, W. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. **Operations Research**, v. 60, n. 3, p. 611–624, 2012.

ZHANG, Y.; MEI, Y.; TANG, K.; JIANG, K. Memetic algorithm with route decomposing for periodic capacitated arc routing problem. **Applied Soft Computing Journal**, p. 17–21, 2016. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.asoc.2016.09.017>>.